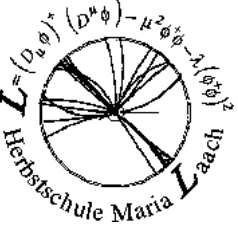
Computing** for High Energy Physics

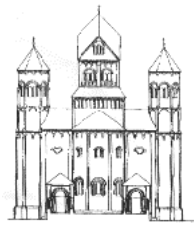
33. Herbstschule für Hochenergiephysik
4.-14.9.2001

Matthias Kasemann
Fermilab

Part 1

***Computing == Computing and Analysis*





In Silica Fertilization

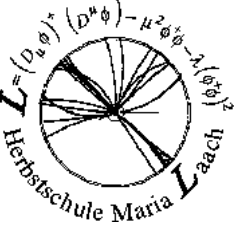
All Science Is Computer Science

By GEORGE JOHNSON

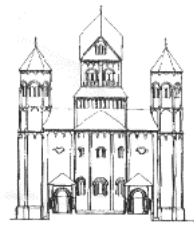
EXCEPT for the fact that everything, including DNA and proteins, is made from quarks, particle physics and biology don't seem to have a lot in common. One science uses mammoth particle accelerators to explore the subatomic world; the other uses petri dishes, centrifuges and other laboratory paraphernalia to study the chemistry of life. But there is one tool both have come to find indispensable: supercomputers powerful enough to sift through piles of data that would crush the unaided mind.

Last month both physicists and biologists made announcements that challenged the tenets of their fields. Though different in every other way, both discoveries relied on the kind of intense computer power that would have been impossible to marshal just a few years ago. In fact, as research on so many fronts is becoming increasingly dependent on computation, all science, it seems, is becoming computer science.

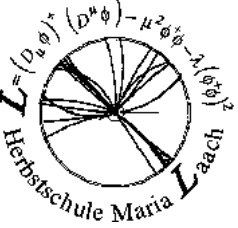
"Physics is almost entirely computational now," said Thomas B. Kepler, vice president for academic affairs at the Santa Fe Institute, a multidisciplinary research center in New Mexico. "Nobody would dream of doing these big accelerator experiments without a tremendous amount of computer power to analyze the data."



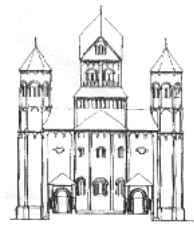
My goals for this lectures



- To present the computing requirements for HEP experiments
- To present tools used for HEP computing
- To convince you:
  - that computing for HEP is as important as building the detector
  - that computing must be planned and 'projectized'
  - that computing cannot be done by physicist alone
  - that computing cannot be done by "professionals" alone
  - work on software and computing must start early
- There is or must/should be a place in physics for building HEP computing and analysis systems as well as building detectors



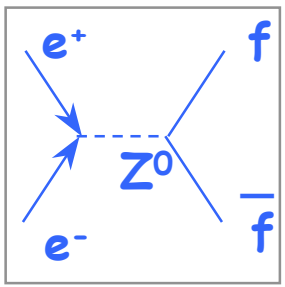
Outline

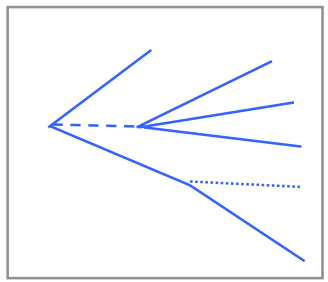


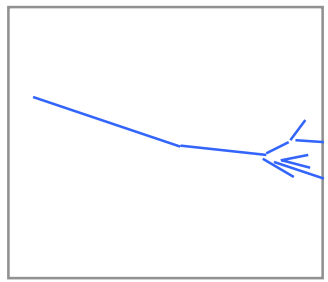
- Computing for HEP experiments
- Existing experiments: choices made and experience
  - BaBar at SLAC
  - D0 (or CDF) at FNAL
- Computing for planned experiments
  - LHC: CMS (or ATLAS)
  - Experiments Software and Computing Projects
  - Distributed, worldwide analysis
- International projects: GRID projects

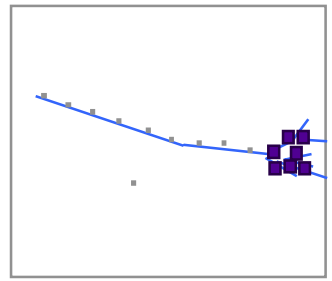
- Specialized computing for HEP: Lattice QCD machines
- Role of Networking

From Physics to Raw Data: what happens in a detector









250Kb - 1 Mb

2037 2446 1733 1699
4003 3611 952 1328
2132 1870 2093 3271
4732 1102 2491 3216
2421 1211 2319 2133
3451 1942 1121 3429
3742 1288 2343 7142

Theoretical Model of Particle interaction

Fragmentation, Decay

Interaction with detector material
Multiple scattering, interactions

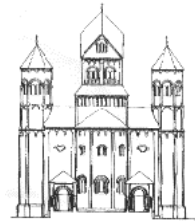
Detector response
Noise, pile-up, cross-talk, inefficiency, ambiguity, resolution, response function, alignment, temperature

Raw data (Bytes)
Read-out addresses, ADC, TDC values, Bit patterns

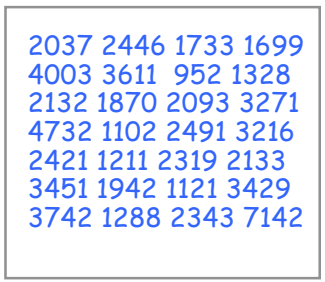
Particle production and decays observed in detectors are Quantum Mechanical processes. Hundreds or thousands of different production- and decay-channels possible, all with different probabilities.

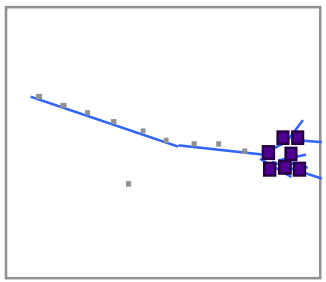
In the end all we measure are probabilities!!

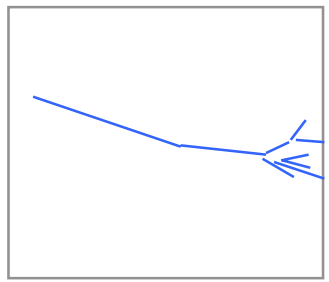
From Raw Data to Physics: what happens during analysis

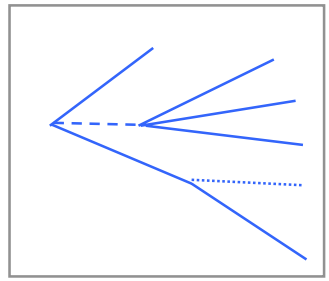


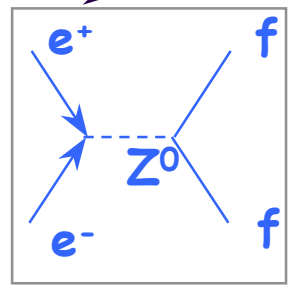
250Kb – 1 Mb → 100 Kb → 25 Kb → 5 Kb → 500 b











Raw data

Convert to physics quantities

Detector response

apply calibration, alignment,

Interaction with detector material

Pattern, recognition, Particle identification

Fragmentation, Decay

Physics analysis

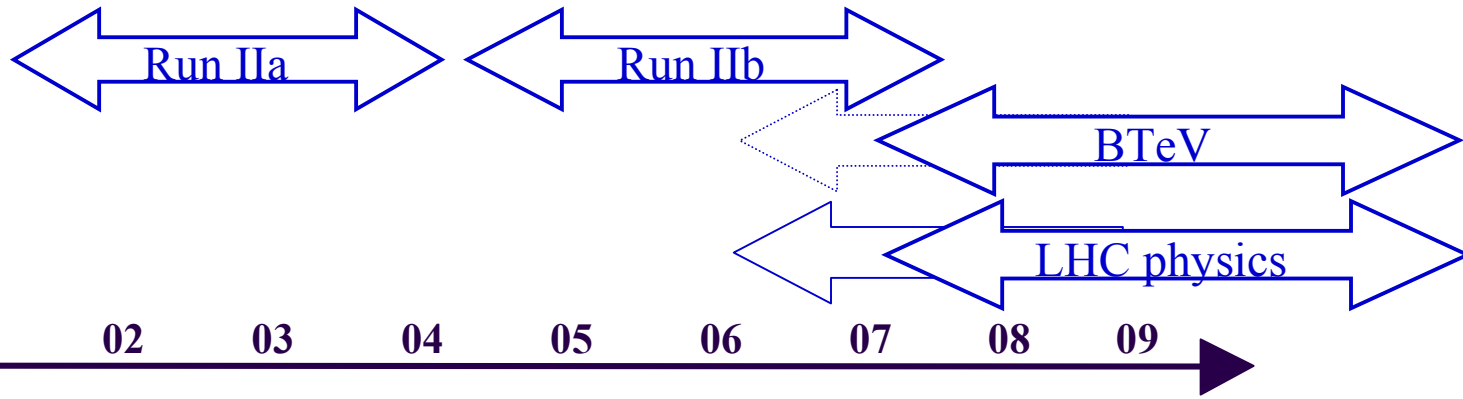
Basic physics

Results

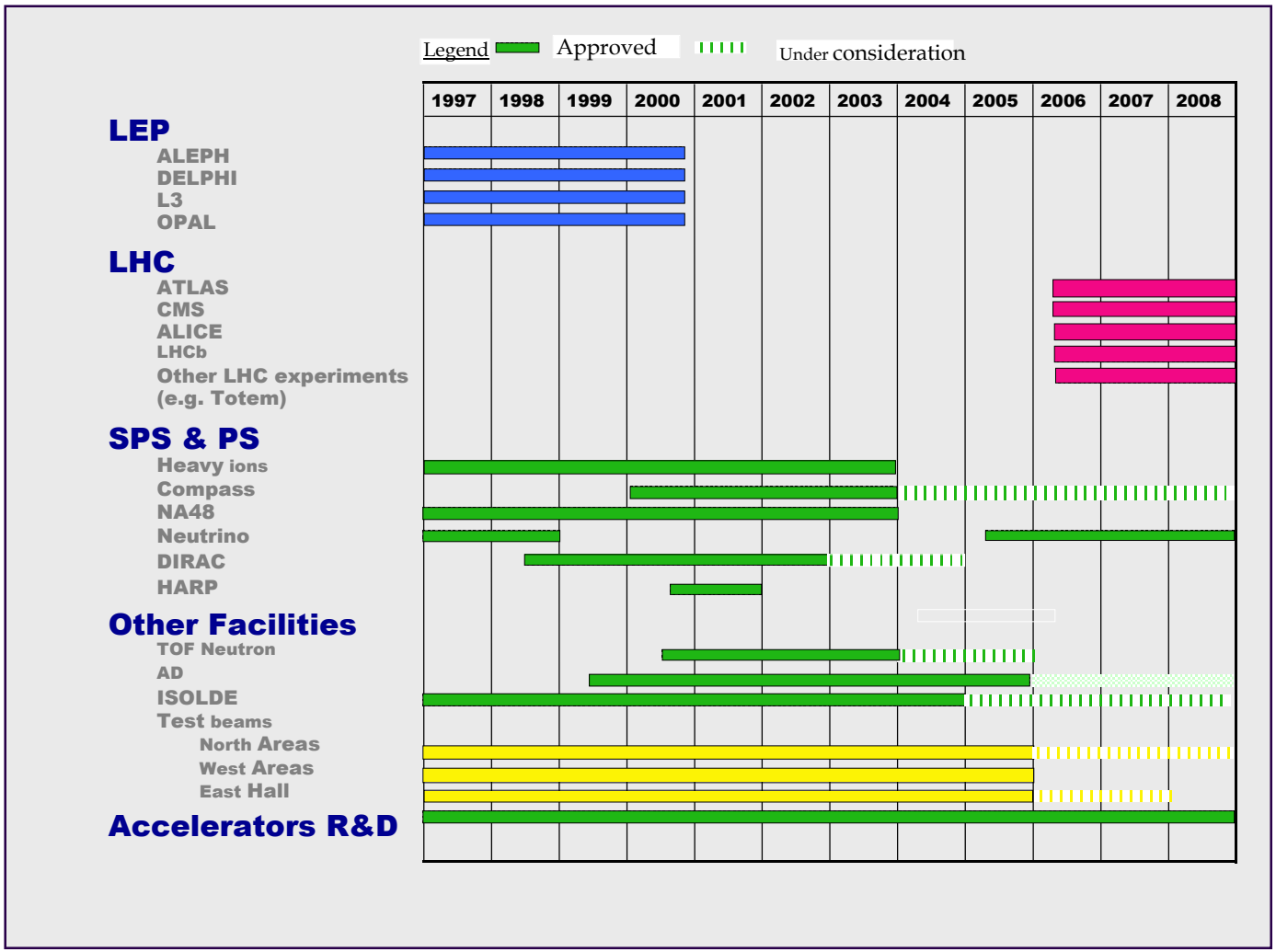
Reconstruction → Analysis

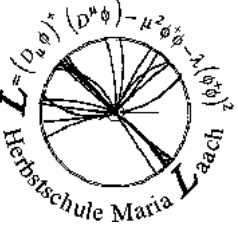
Simulation (Monte-Carlo)

Fermilab HEP Program

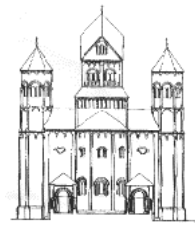


The CERN Scientific Programme

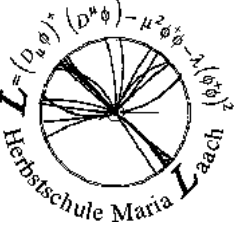




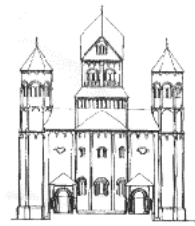
Past, Present and Future - define the terms



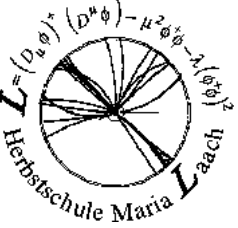
- Past -- Solutions already implemented
  - in use today - HEP expts, SloanDigitalSky Survey,Theorist Lattice Guage Computation
  - operational experience (e.g. with HPSS)
- Present -- Solutions being implemented for Collider Run II with upgraded detectors (March 2000)
  - Joint Computing Project for CDF and D0 Run II Computing
  - Prototyping and testing data handling solutions
- Future -- Participation in longer term futures for LHC and other future experiments
  - CMS experiment
  - HPSS, DESY/Eurostore
  - MONARC
  - Databases exploration



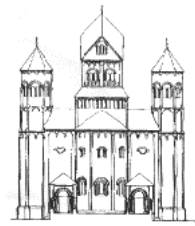
HEP computing: The next 5 years…(1)



- Data analysis for completed experiments continues
  - Challenges:
    - No major change to analysis model, code or infrastructure
    - Operation, continuity, maintaining expertise and effort
- Data collection and analysis for ongoing experiments
  - Challenges:
    - Data volume, compute resources, software organization
    - Operation, continuity, maintaining expertise and effort

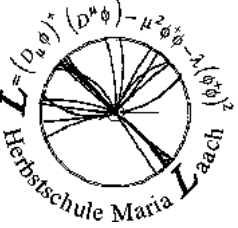


HEP computing: The next 5 years…(2)

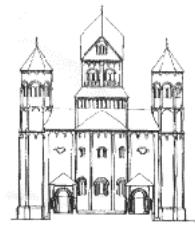


- Starting experiments:
  - Challenges:
    - Completion and verification of data and analysis model,
    - Data volume, compute resources, software organization, $$'s
    - Operation, continuity, maintaining expertise and effort

- Experiments in preparation:
  - Challenges:
    - Definition and implementation of data and analysis model,
    - data volume, compute resources, software organization, $$'s
    - continuity, getting and maintaining expertise and effort
    - Build for change: applications, data models…
    - Build compute models which are adaptable to different local environments

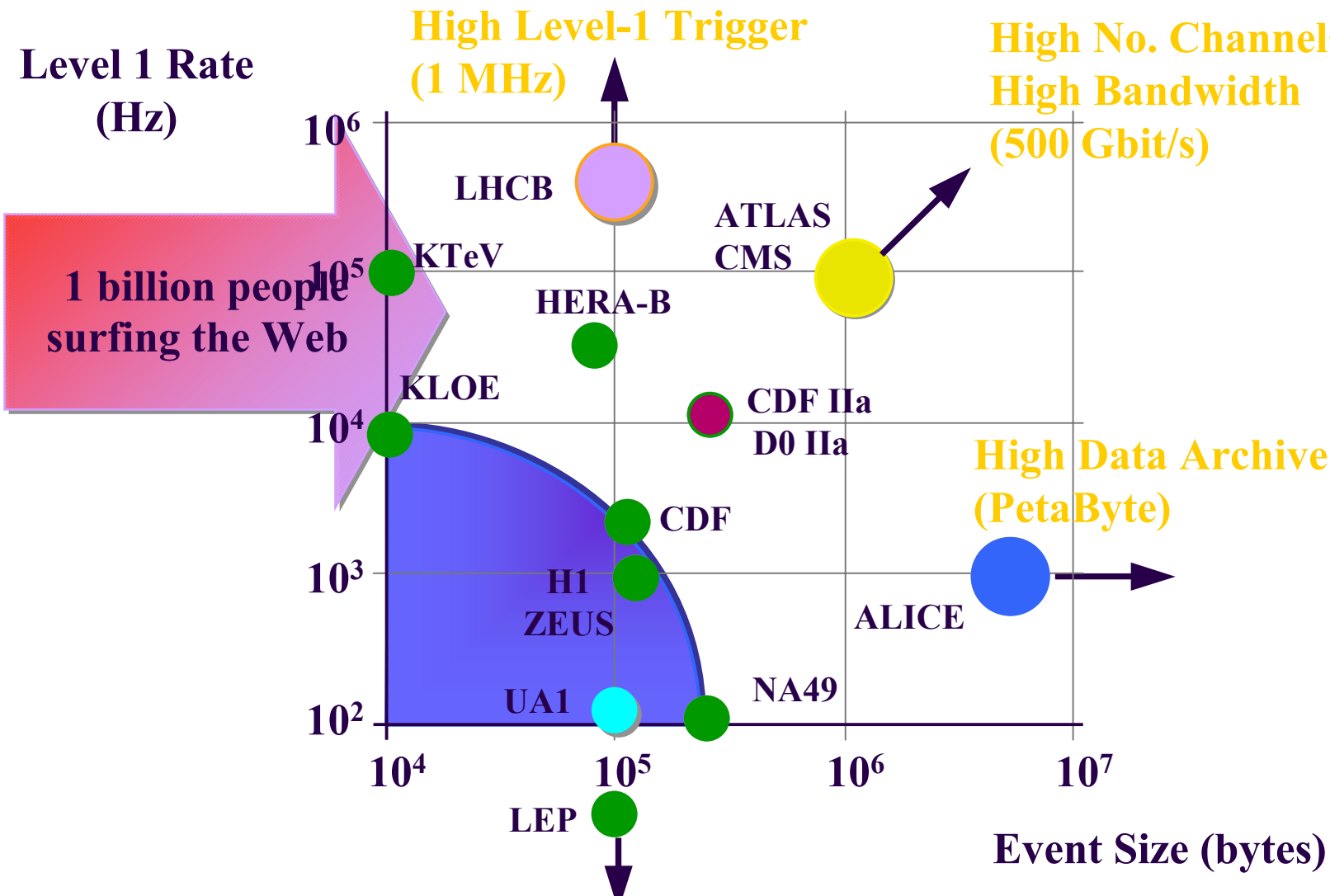


HEP computing: The next 5 years…(3)

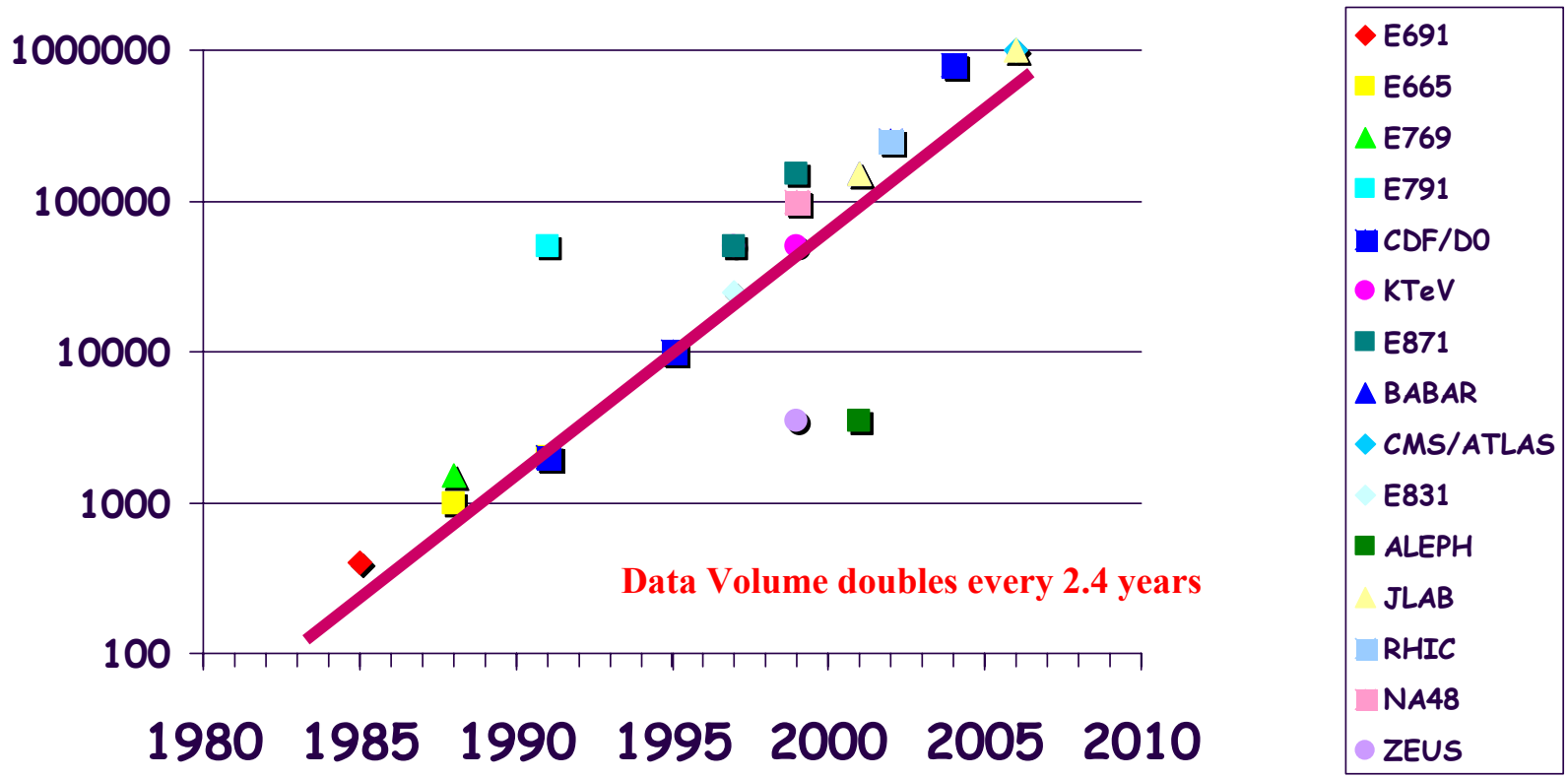


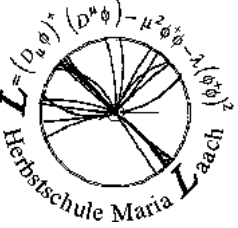
- Challenges in big collaborations
  - Long and difficult planning process
  - More formal procedure required to commit resources
  - Long lifetime, need flexible solutions which allow for change
    - Any state of experiment longer than typical Ph.D. or postdoc time
    - Need for professional IT participation and support
- Challenges in smaller collaborations
  - Limited in resources
  - Adapt and implement available solutions (“b-b-s”)

How Much Data is Involved?

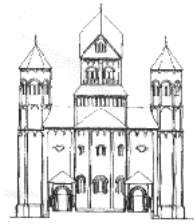


Data Volume per experiment per year (in units of Gbytes)

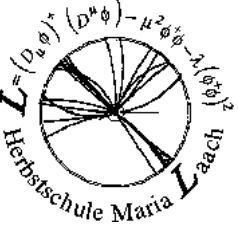




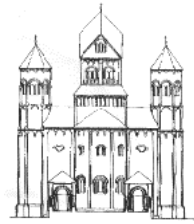
How long are data scientifically interesting? ("Lifetime of data")



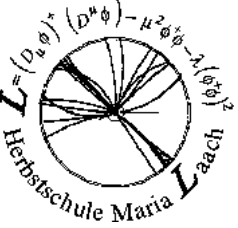
- 1. Month after recording:
  - Verification of data integrity
  - Verification of detector performance and integrity
- 6-12-24 months after recording:
  - Collect more data
  - Process and reconstruct "interpret the bits"
  - Perform data analysis
  - Compare to simulated data
  - Publish!!
- >2 years after recording:
  - Data often superseded by more precise experiments
  - Combine results for high statistics measurements and publish!!
  - Archive for comparison and possible re-analysis
- >5 years after recording:
  - Decide on long-term storage for re-analysis



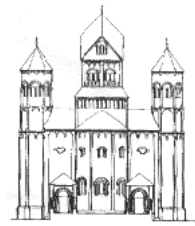
Data analysis in international collaborations: past



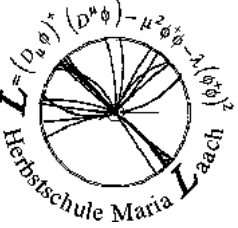
- In the past analysis was centered at the experimental sites
  - a few major external centers were used.
  - Up the mid 90s bulk data were transferred by shipping tapes, networks were used for programs and conditions data.
  - External analysis centers served the local/national users only.
  - Often staff (and equipment) from the external center being placed at the experimental site to ensure the flow of tapes.
  - The external analysis often was significantly disconnected from the collaboration mainstream.



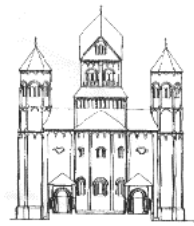
Data analysis in international collaborations: truly distributed



- Why?
  - For one experiment looking ahead for a few years only centralized resources may be most cost effective, but:
  - national and local interests leads to massive national and local investments
  - For BaBar:
    - The total annual value of foreign centers to the US-based program is greatly in excess of the estimated cost to the US of creating the required high-speed paths from SLAC to the landing points of lines WAN funded by foreign collaborators
  - Future world-scale experimental programs must be planned with explicit support for a collaborative environment that allows many nations to be full participants in the challenges of data analysis.

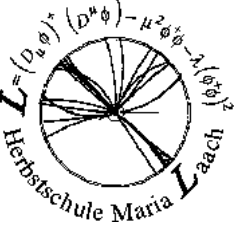


Computing Needs: Comparison between Experiments

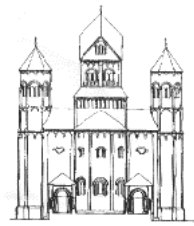


Experiment	Onsite CPU (SI95)	Onsite Disk (TB)	Onsite Tape (TB)	LAN Capacity	Data import/ export	Box count
500MHz PIII	20					
CMS	520,000	540	2000	46 GB/s	10 TB/day	~1400
CDF(Run2)	12,000	20	800	?		~250
D0(Run 2)	7,000	20	600	300 MB/s		~250
BaBar	10,000	10	300		0.5 TB/day	~300
CDF(Run1)	280	?	?	?		?
D0(Run 1)	295	1.5	65	300 Mb/s		180
ALEPH	300	?	5.5	1 Gb/s		1
DELPHI	515	1.2	?	1 Gb/s		20
L3	625	2	?	1 Gb/s	none	1
OPAL	835	1.6	?	1 Gb/s		1
NA45	587	1.3	2	1 Gb/s	5 GB/day	30
NA48	650	4.5	140	1 Gb/s	5 GB/day	50
KTeV	280	1	50	100 Mb/s	150 GB/day	2

Status: as of 1999



Distributed computing:



- Networking is an expensive resource, should be minimized
- Pre-emptive transfers can be used to improve responsiveness at the cost of some extra network traffic.
- Multi-tiered architecture must become more general and flexible
  - to accommodate the very large uncertainties in the relative costs of CPU, storage and networking
  - To enable physicists to work effectively in the face of data having unprecedented volume and complexity
- Aim for transparency and location independence of data access
  - the need for individual physicists to understand and manipulate all the underlying transport and task-management systems would be too complex

Distributed Computing

6/13/01: The New York Times ON THE WEB

"It turns out that distributed computing is really hard,"

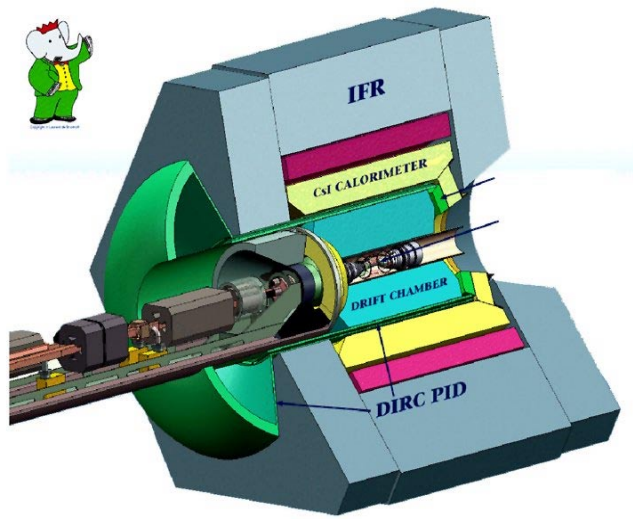
said Eric Schmidt, the **chairman of Google**, the Internet search engine company.

"It's much harder than it looks. It has to work across different networks with different kinds of security, or otherwise it ends up being a single-vendor solution, which is not what the industry wants."

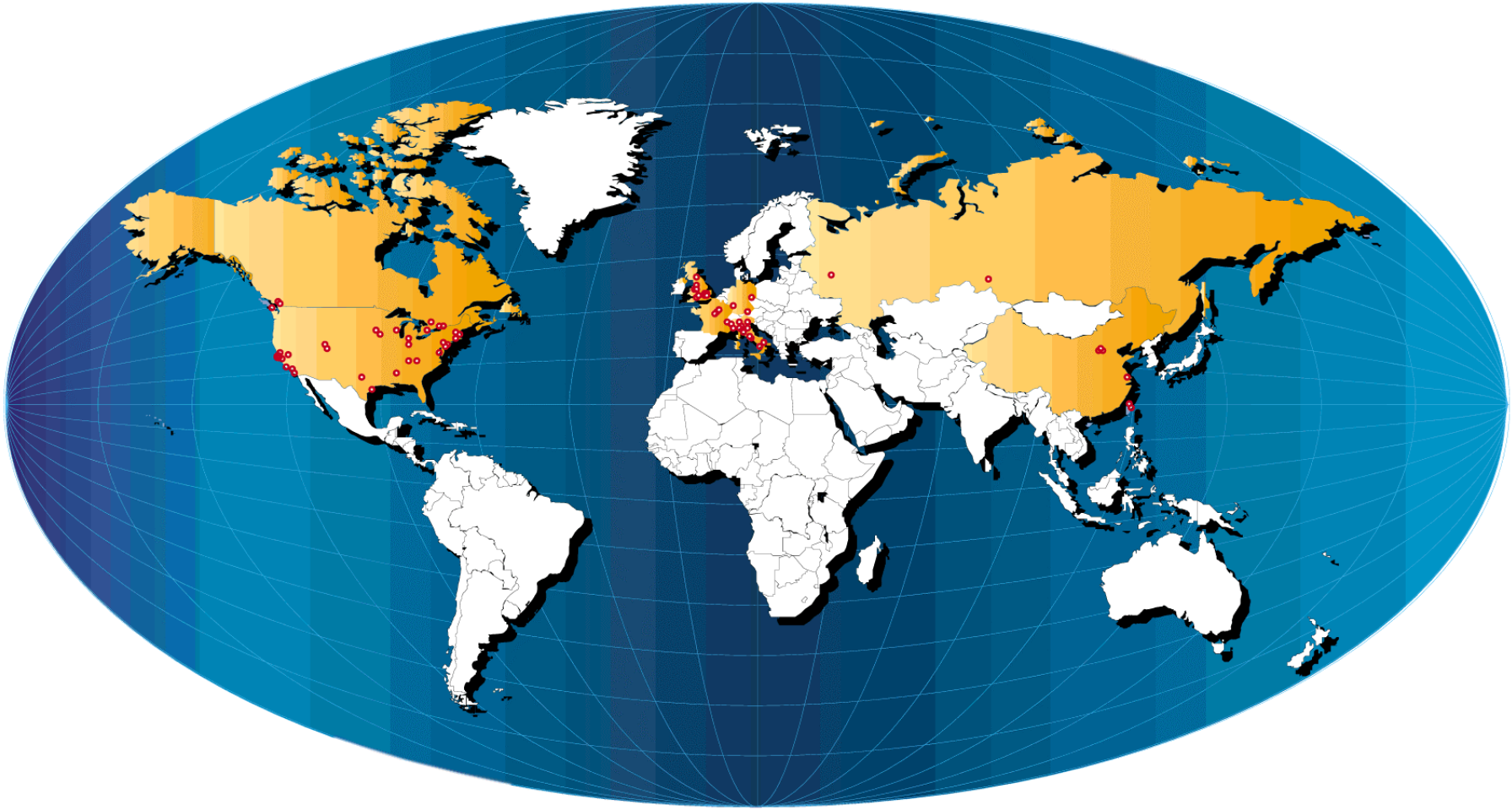
Google™

BaBar: Statistics

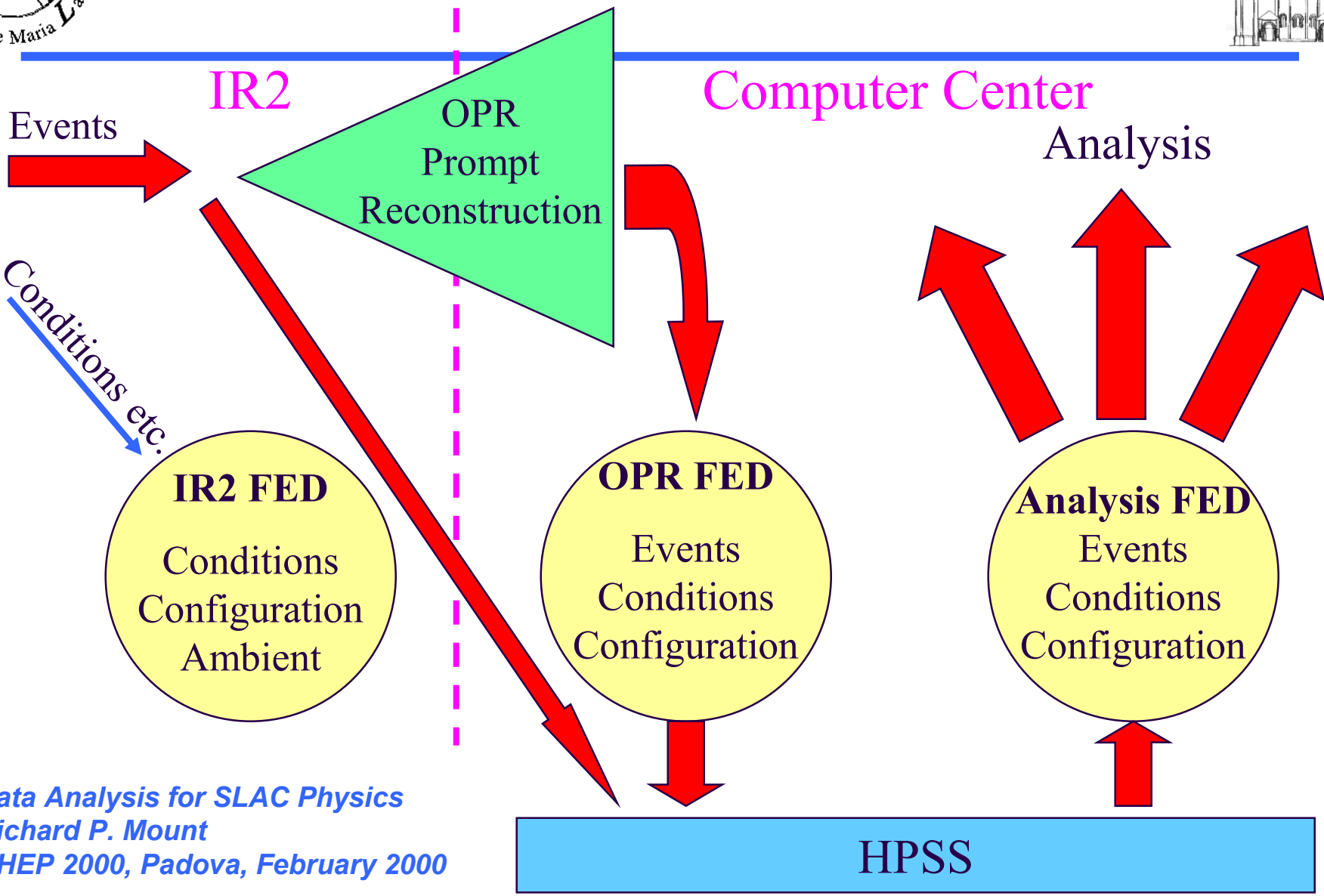
- CP B physics requires a 30 fb-1 yearly sample for > 5 years
  - One year is 30M B events, 120M hadronic, 1.2B Bhabhas seen
  - “Factory mode” running for greater than 80% of real time
  - 100Hz of accepted L3 triggers to be read, 30Hz to fully process
    - Roughly 3MB/sec of raw data, all day, every day
    - Similar size downstream processing and analysis streams
- High capability detector
  - 5 layer Silicon Vertex tracker
  - 40 layer low mass drift chamber
  - Novel “DIRC” particle ID
  - Crystal calorimeter
  - Highly segmented instrumented flux return
- But life is never easy
  - Severe machine backgrounds
  - Significant compromises in geometry & regularity



BaBar: Worldwide Collaboration of 80 Institutes

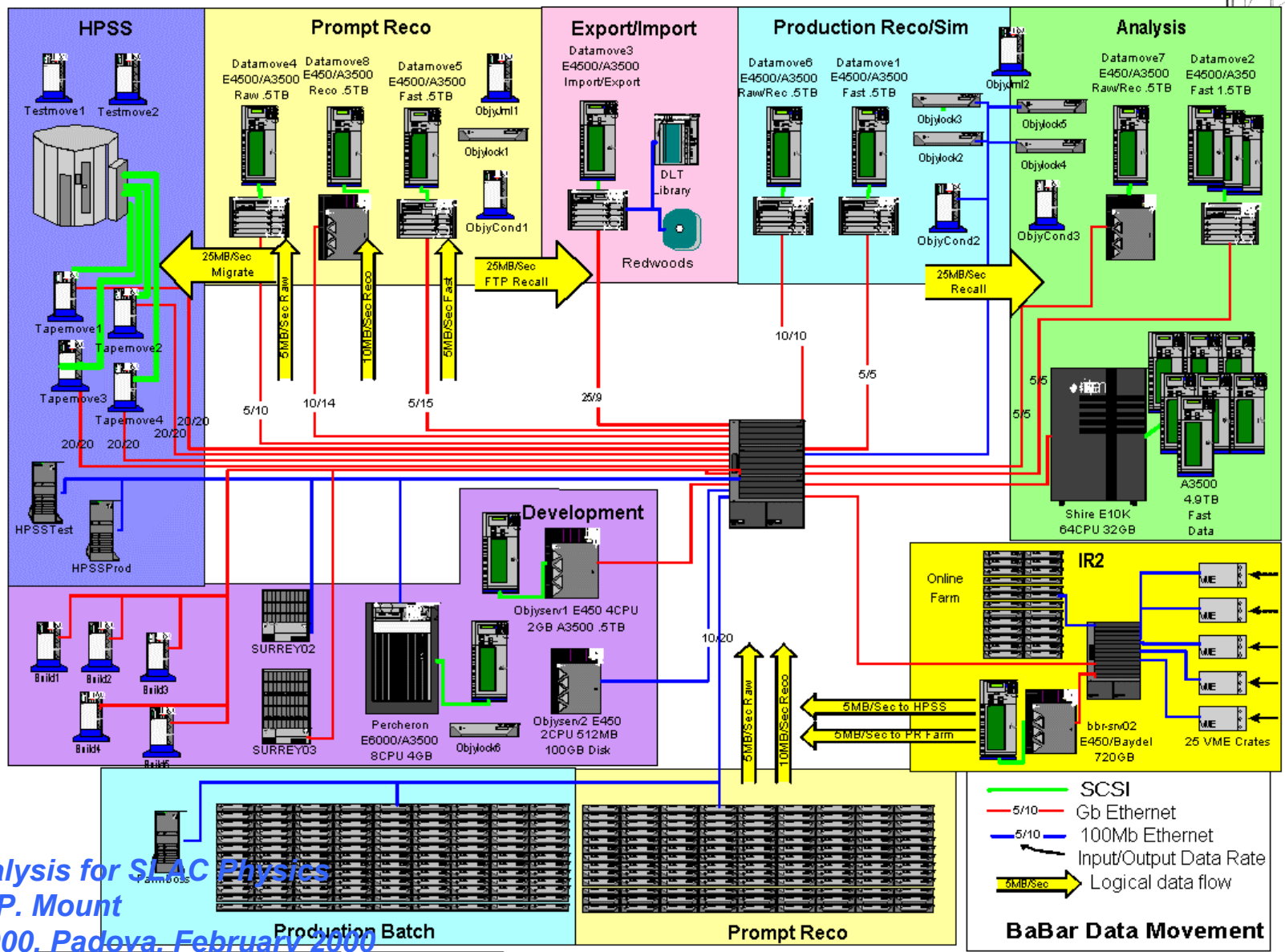


BaBar: Principal Data Flows

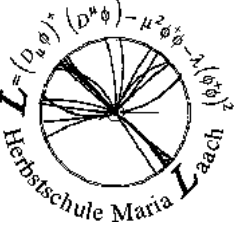


Data Analysis for SLAC Physics
Richard P. Mount
CHEP 2000, Padova, February 2000

BaBar Offline Systems: August 1999

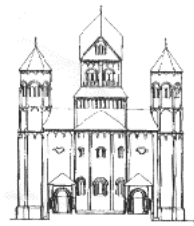


Data Analysis for SLAC Physics
Richard P. Mount
CHEP 2000, Padova, February 2000

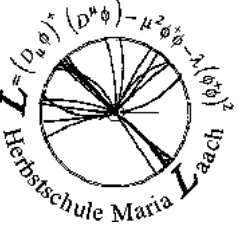


SLAC-BaBar Data Analysis System

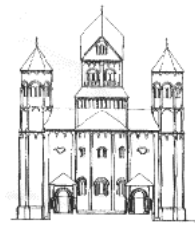
50/400 simultaneous/total physicists, 300 Tbytes per year



HARDWARE	UNITS	End FY1999	End FY2000
Tape Silos (STK Powderhorn, 6000 tapes each)	silos	6	6
Tape Drives (STK Eagle, 20 Gbyte, 10 Mbytes/s)	drives	20	40
Disk (net capacity of RAID arrays)	Tbytes	20	56
File Servers and Data Movers (Sun)	CPUs	73	150
Interactive Servers (Sun + Linux)	CPUs	82	140
Batch Servers (Sun + Linux)	CPUs	300	900
Network Switches (Cisco 6509)	switches	5	14

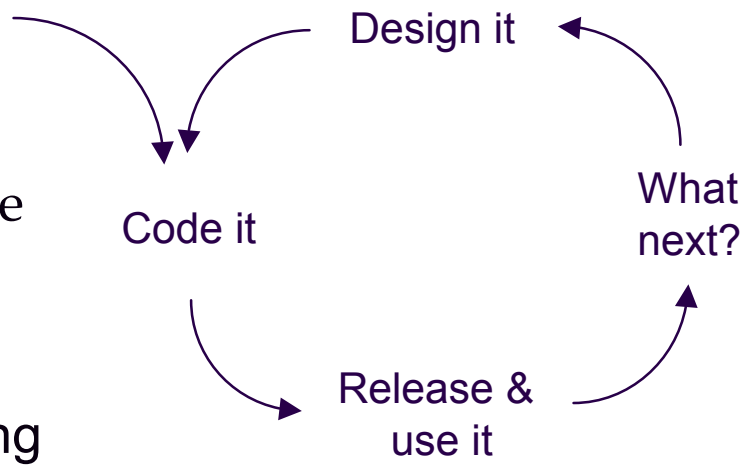


BaBar: Our "design process"



What has BaBar learned
Bob Jacobsen
CHEP 2000, Padova, February 2000

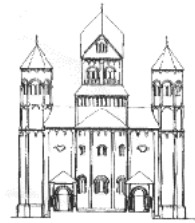
- We use an evolutionary approach
  - People enter coding
  - Eventually, they start to draw clouds and blobs
  - Many of them become good designers
- Evolution improves the system
  - Relevant code is used
    - Comments are not always gentle
  - Release system controls the pace
    - Biweekly timescale
  - New designs, redesigns are ongoing
    - Driven by perceived needs



- Policy: "Get them engaged, then work with them"

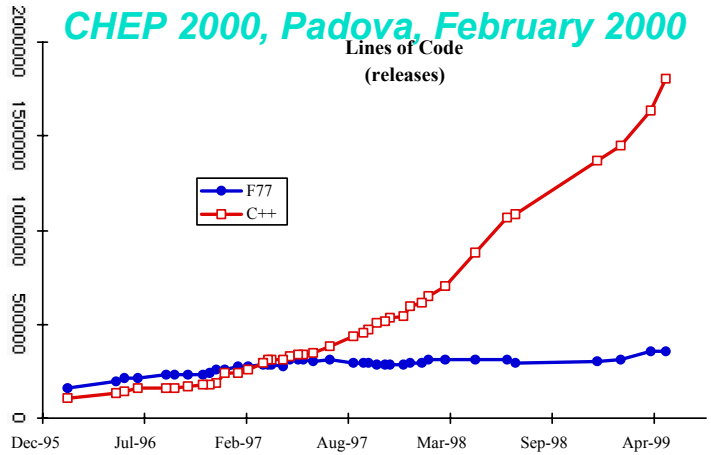
CHEP97 slide

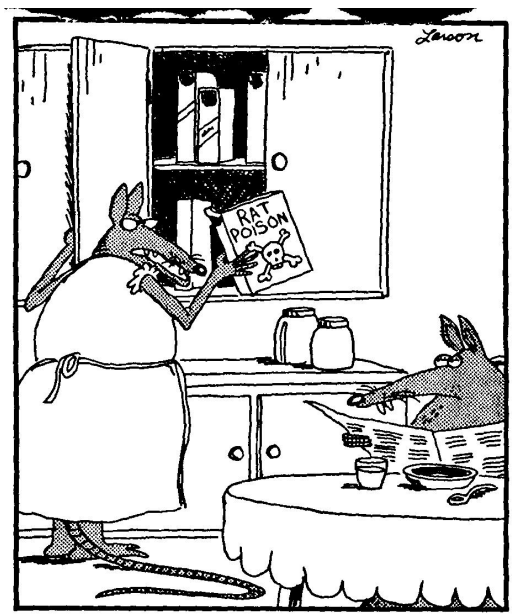
C++ & OO



What has BaBar learned
Bob Jacobsen
CHEP 2000, Padova, February 2000

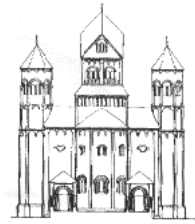
- People will write C++
- Structure varies a lot
  - FORTRAN with ; and #include
  - C with abstract datatypes
  - "The True Style" (whatever that means)
    - data hiding
    - reuse by inheritance
    - abstract interfaces
    - generic programming
- Flexibility is both a strength and a weakness
- We've had some very significant successes
  - Calibration model
  - Track model
  - Physics analysis tools





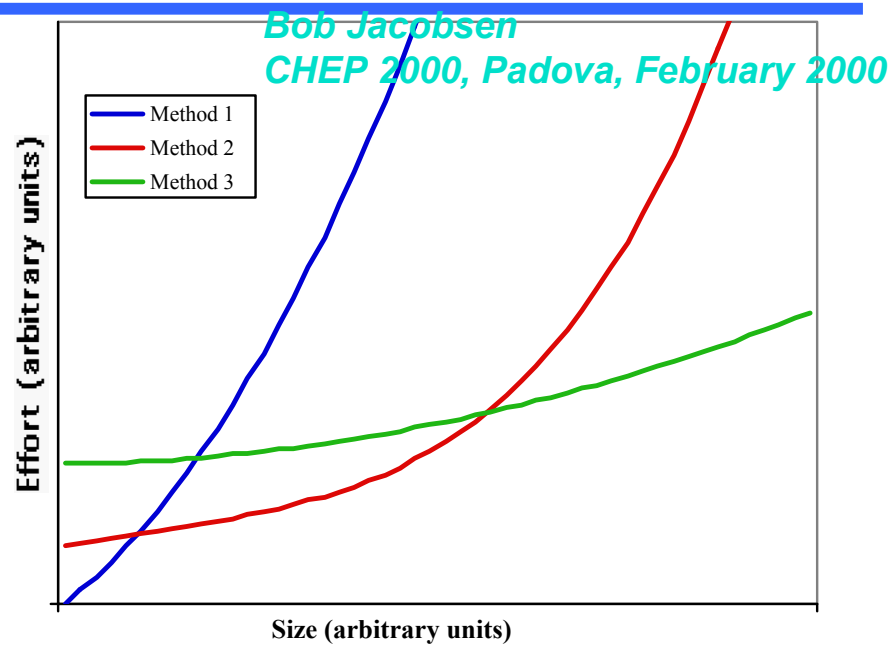
"You know, it's really dumb to keep this right next to the cereal. ... In fact, I don't know why we even keep this stuff around in the first place."

"C++ is harder to learn than FORTRAN"

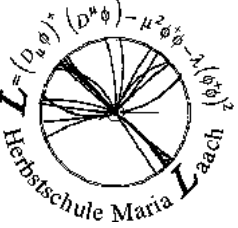


What has BaBar learned
Bob Jacobsen
CHEP 2000, Padova, February 2000

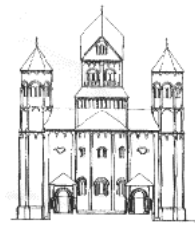
- Unfortunate, but true
  - Perhaps you can justify it
  - Can lead to mistakes
- Need efforts to limit impact
  - Training
  - Mentoring



- C++ and especially OO puts off a number of senior, experienced people
  - Even with specific efforts to couple in, this has cost us
  - PI's less likely than postdocs to contribute to reconstruction and simulation
  - Will it extend to analysis? For how long?



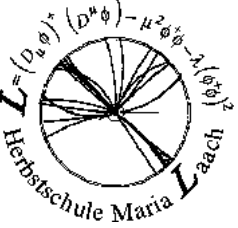
BaBar: The concept of software project management



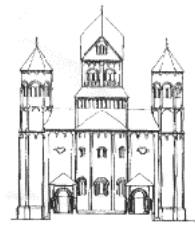
What has BaBar learned
Bob Jacobsen
CHEP 2000, Padova, February 2000

- **Something new in this generation of experiments**
  - Bigger systems are possible/necessary now, and they sop up all the technical gains
  - Example: BaBar analysis tools run in production
- **Still learning how to do this**
  - Similar, yet different from hardware projects
  - BaBar's matrix organization by system and computing area
    - Which way will people sign up in the beginning?
    - Which is more stable in the long run?
- **"Data handling" as respected subject**
  - Collaborations paying attention in advance
  - Bookkeeping critical to success
  - Robots allow access to raw data, instead of waiting for yearly bulk reprocessing
  - Big issue for off-site work - is this really getting better?

"In art, intentions are not enough. What counts is what one does, not what one intends to do." - Pablo Picasso



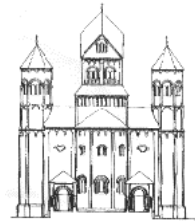
Code Management & Release Management



What has BaBar learned
Bob Jacobsen
CHEP 2000, Padova, February 2000

- CVS - SoftRelTools approach
  - Collaboration-wide read/write access to code in CVS
    - Organized as 630 packages
    - We don't attempt to keep the HEAD production quality
  - Package coordinators
    - One per package
    - Tags and announces when new version ready for use
  - Build periodic releases from these tagged versions
    - Integration and testing to production now takes two weeks
- "100KLOC is easy; we know how to do 1MLOC; 10MLOC is hard"
  - Examples of what we've had trouble with
    - Transition to "use & production" instead of development
    - Introduced a more reliable (rigid) one month cycle
    - Imposing a freeze on processing code now to create summer CP violation sample
  - Cannot imagine getting this "right"
- New issue - runtime environment management

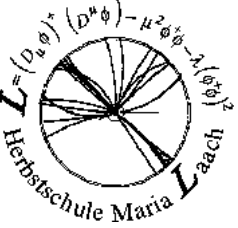
Distributed multi-platform development



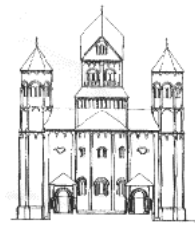
What has BaBar learned
Bob Jacobsen
CHEP 2000, Padova, February 2000

- Use native compilers & tools on Sun/Solaris, DEC/Compaq and Linux
  - Code to a common subset, empirically enforced
- We're still waiting for the compiler promised land
  - Recent migration to Linux was interesting
  - Still need to think about issues beyond C++ semantics/syntax
    - E.g. template instantiation, inline tricks
  - Ongoing problems with STL, bool
- Complete builds take days
  - Especially with optimization
  - Poor interactions with templates
- People keep saying compilers are getting better. It is not happening fast.

"I've got it, too, Omar ... a strange feeling like we've just been going in circles."

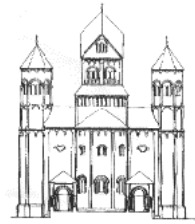


Kanga, the BaBar "Objectivity-Free" Root-I/O-based Alternative



- Aimed at final stages of data analysis
- Easy for universities to install
- Supports BaBar analysis framework
- Very successful validation of the insulating power of the BaBar transient-persistent interface
- Nearly working

Exporting the Data



- CCIN2P3 (France)
  - Plan to mirror (almost) all BaBar data
  - Currently have “Fast” (DST) data only (~3 TB)
  - Typical delay is one month
  - Using Objectivity
- CASPUR (Italy)
  - Plan only to store “Fast” data (but its too big)
  - Data are at CASPUR but not yet available
  - Prefer Kanga
- RAL (UK)
  - Plan only to store “Fast” data
  - Using Objectivity

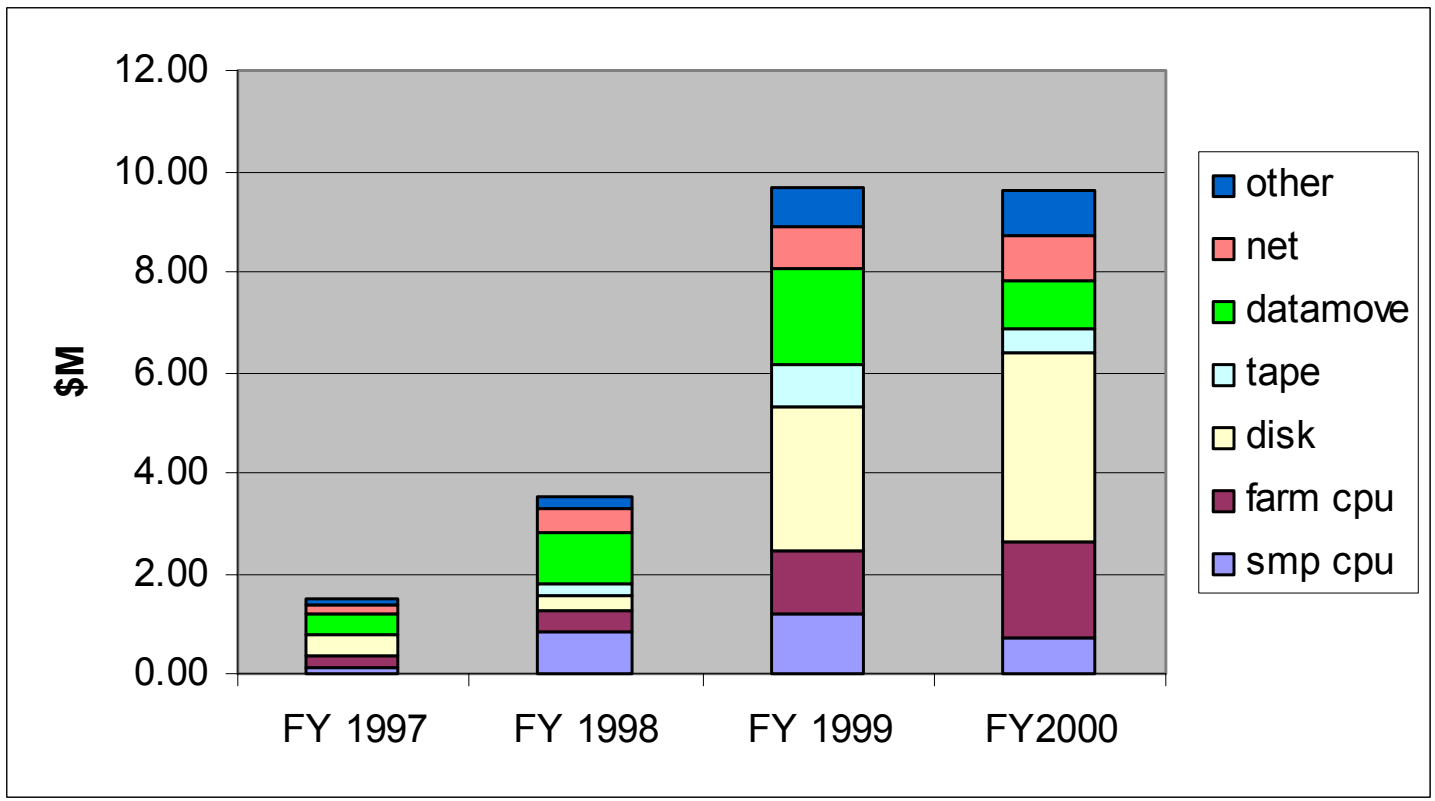
Data Analysis for SLAC Physics
Richard P. Mount
CHEP 2000, Padova, February 2000

BaBar Offline Computing at SLAC:

Costs other than Personnel

(does not include "per physicist" costs such as desktop support, help desk, telephone, general site network)

Data Analysis for SLAC Physics
Richard P. Mount
CHEP 2000, Padova, February 2000



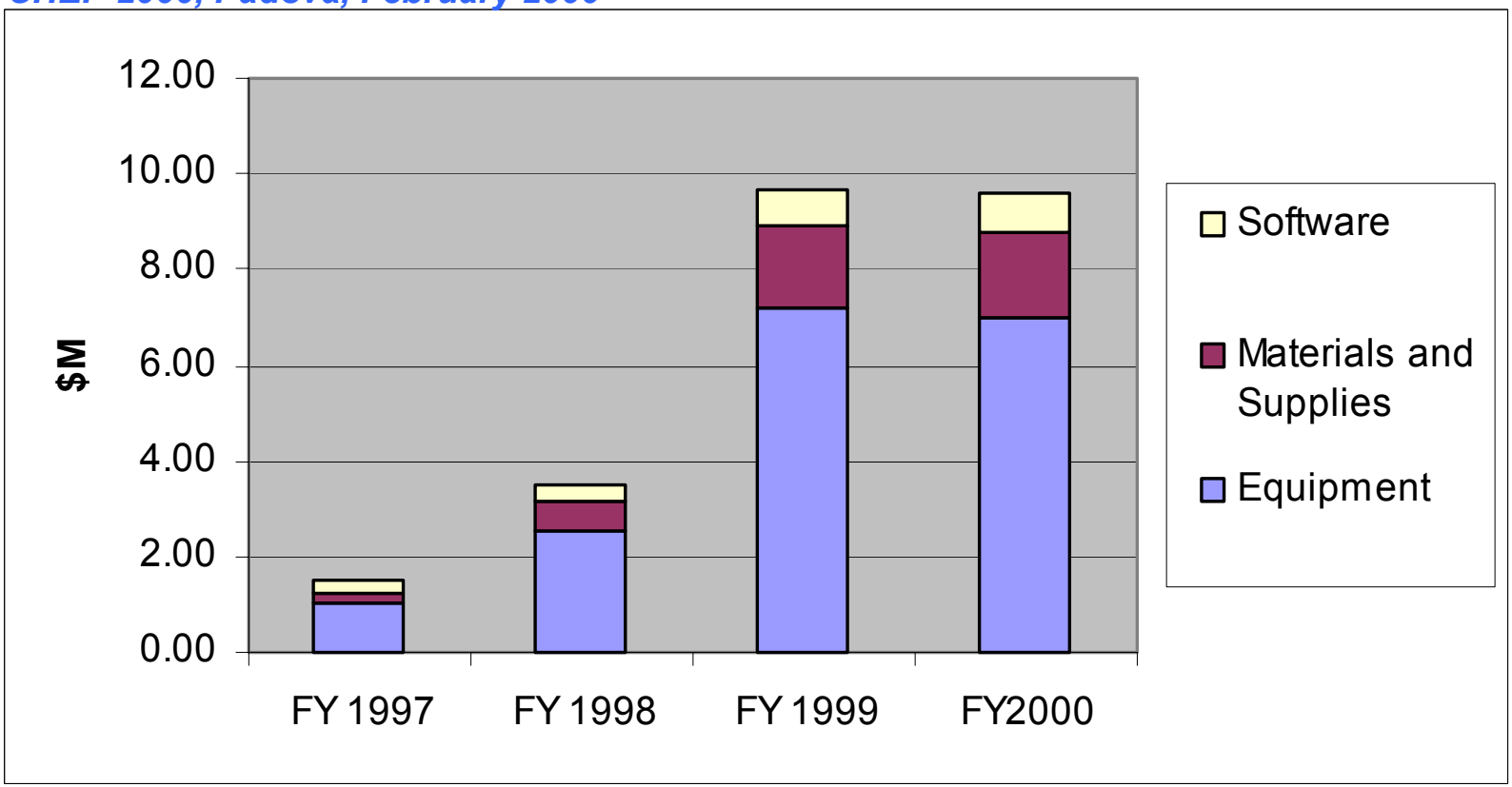
Does not include tapes

BaBar Offline Computing at SLAC:

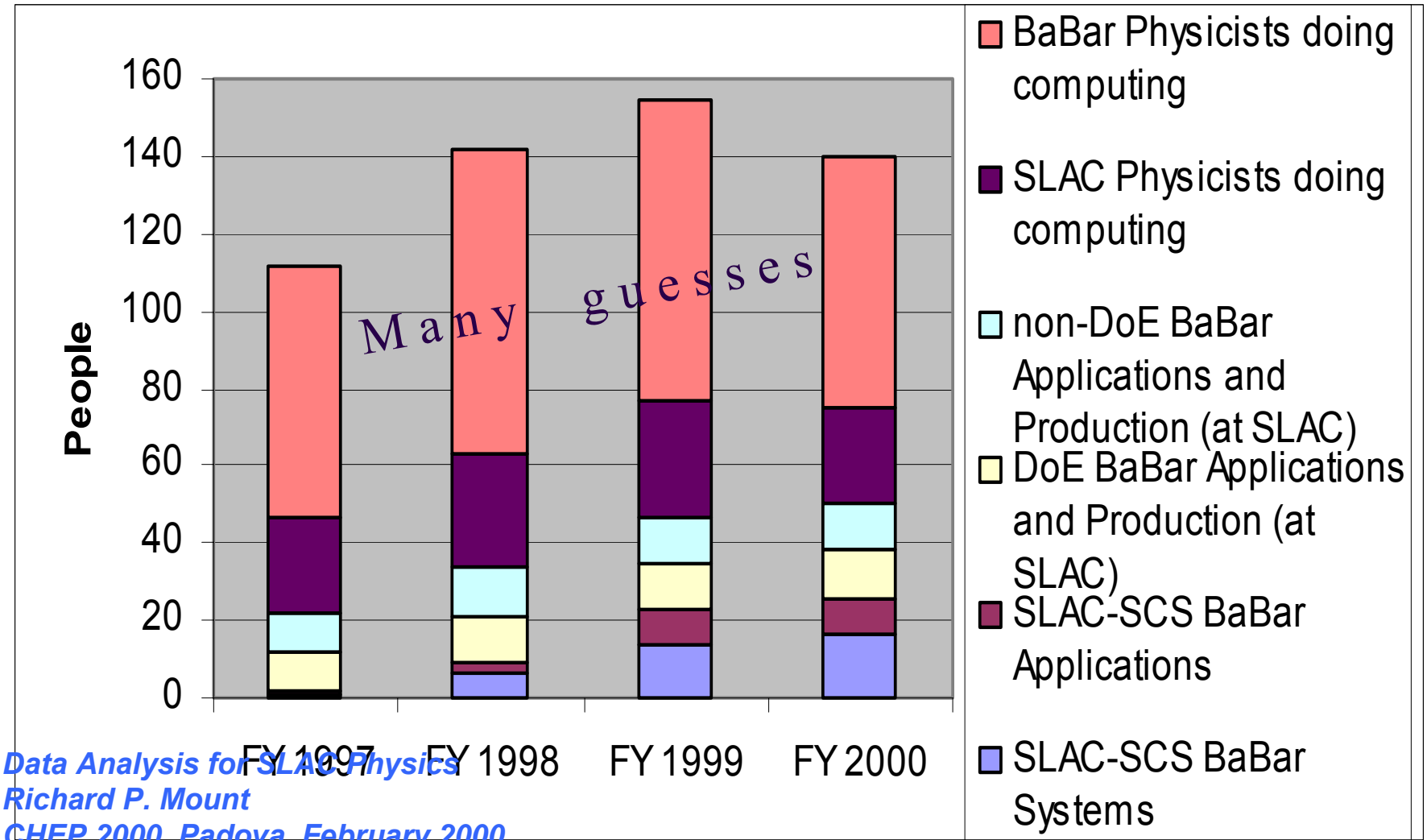
Costs other than Personnel

(does not include "per physicist" costs such as desktop support, help desk, telephone, general site network)

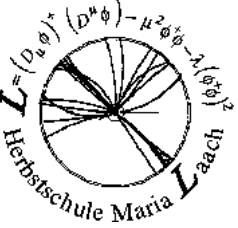
Data Analysis for SLAC Physics
Richard P. Mount
CHEP 2000, Padova, February 2000



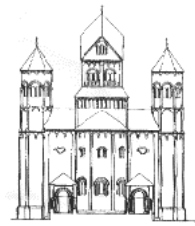
BaBar Computing Personnel The Whole Story?



Data Analysis for SLAC Physics
Richard P. Mount
CHEP 2000, Padova, February 2000

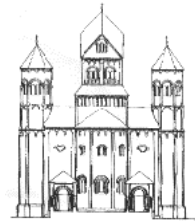


Complexity



- BaBar (and CDF,D0,RHIC,LHC) is driven to systems with ~1000 boxes performing tens of functions
- How to deliver reliable throughput with hundreds of users?
  - Instrument heavily
  - Build huge test systems
  - “Is this a physics experiment or a computer science experiment?”

SLAC: Personnel versus Equipment



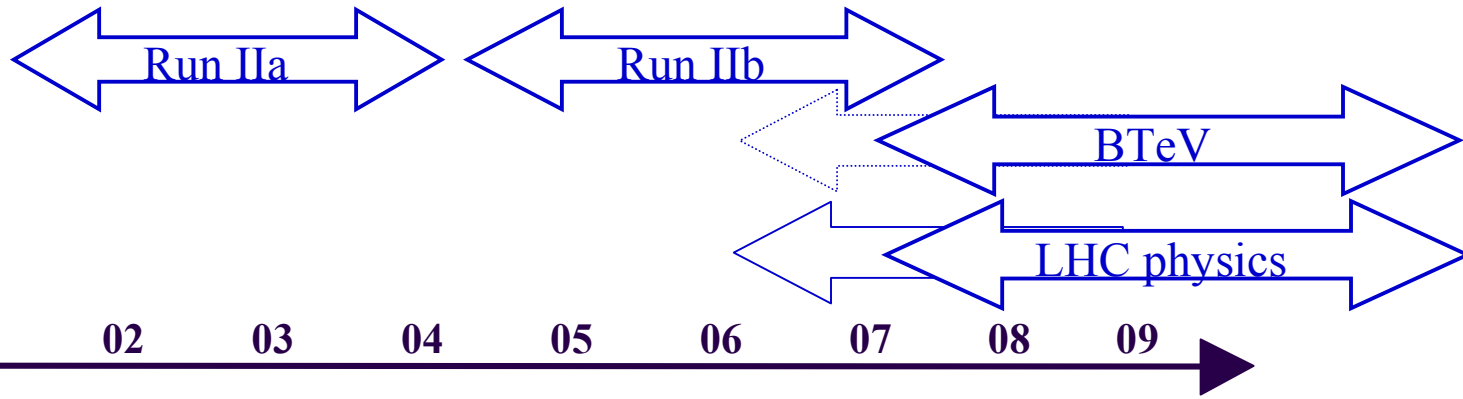
- Should SLAC be spending more on people and buying cheaper stuff?
  We buy:
  - Disks at 5 x rock bottom
  - Tape drives at 5 x rock bottom
  - Farm CPU at 2-3 x rock bottom
  - Small SMP CPU at 2-3 x farms
  - Large SMP CPU at 5-10 x farms
  - Network stuff at “near monopoly” pricing

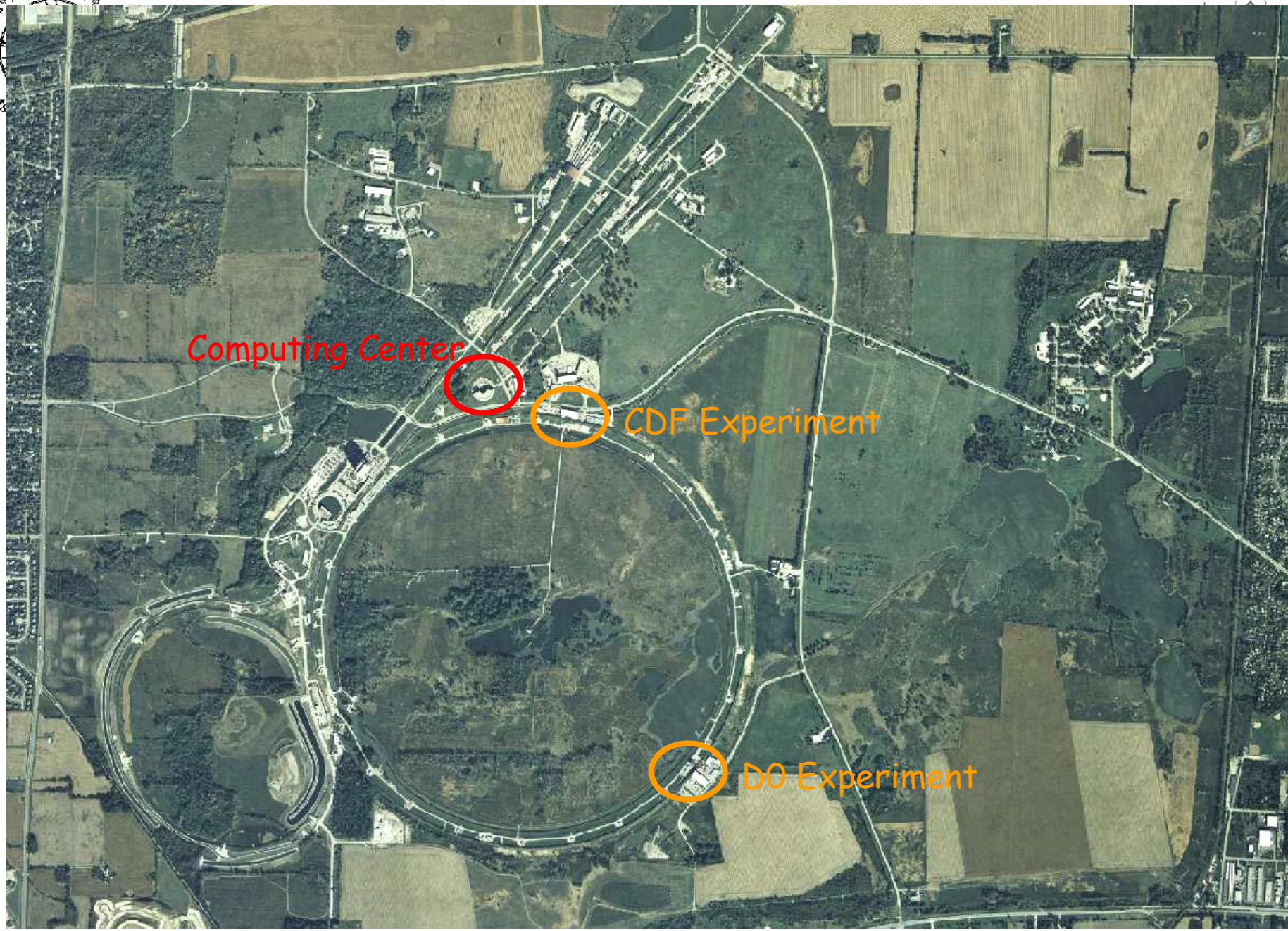
  All at (or slightly after) the very last moment

Richard Mount:

“I am uneasily happy with all these choices”

Fermilab HEP Program



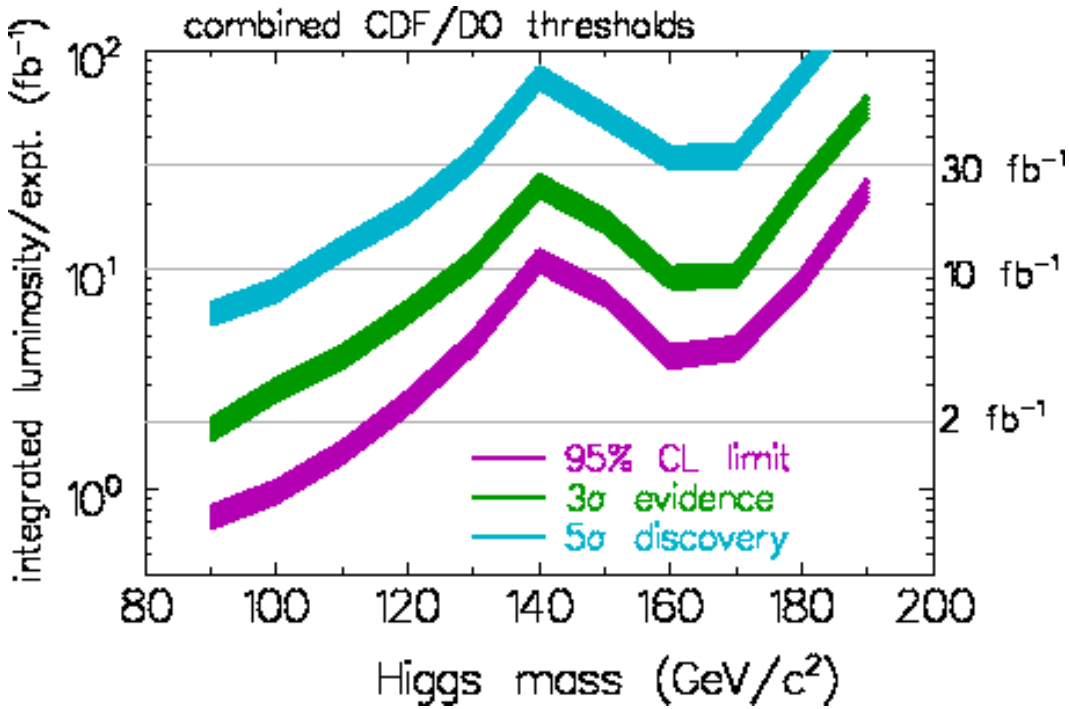

Computing Center
CDF Experiment
DO Experiment

"Run 2 Science: The hunt is on"

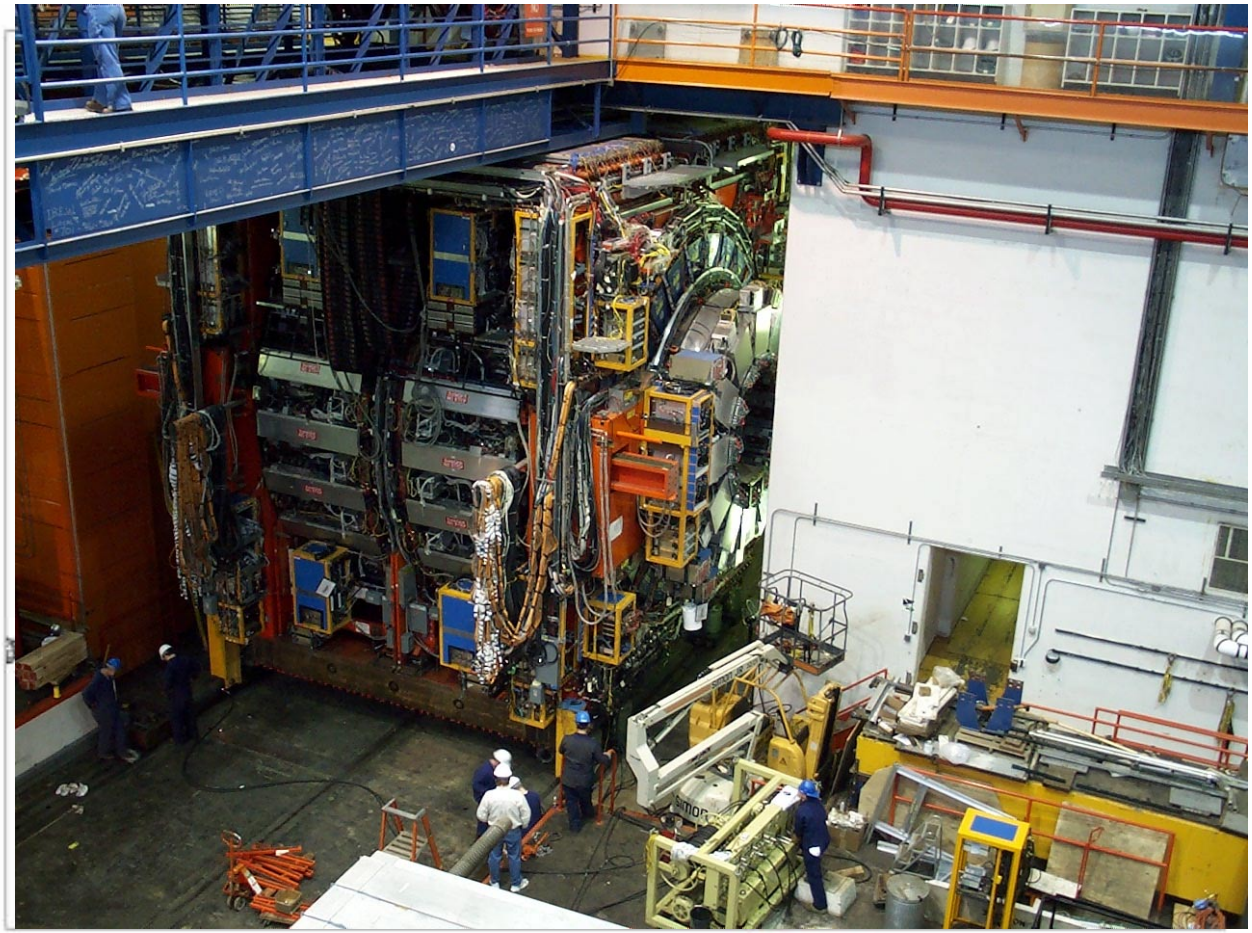
FermiNews, Vol 24, March 2, 2001

- Run 2: Commissioning began Nov. 2000/March 2001 - start of the Fermilab Tevatron Collider Run 2
  - Further upgrades will continue increasing the luminosity. They require a **big** effort.

Run	Dates	Integrated Lumi (fb^{-1})
I	1993-1996	0.1
IIa	2001-2003	~2
IIb	2004-2007	~15



The upgraded CDF Detector

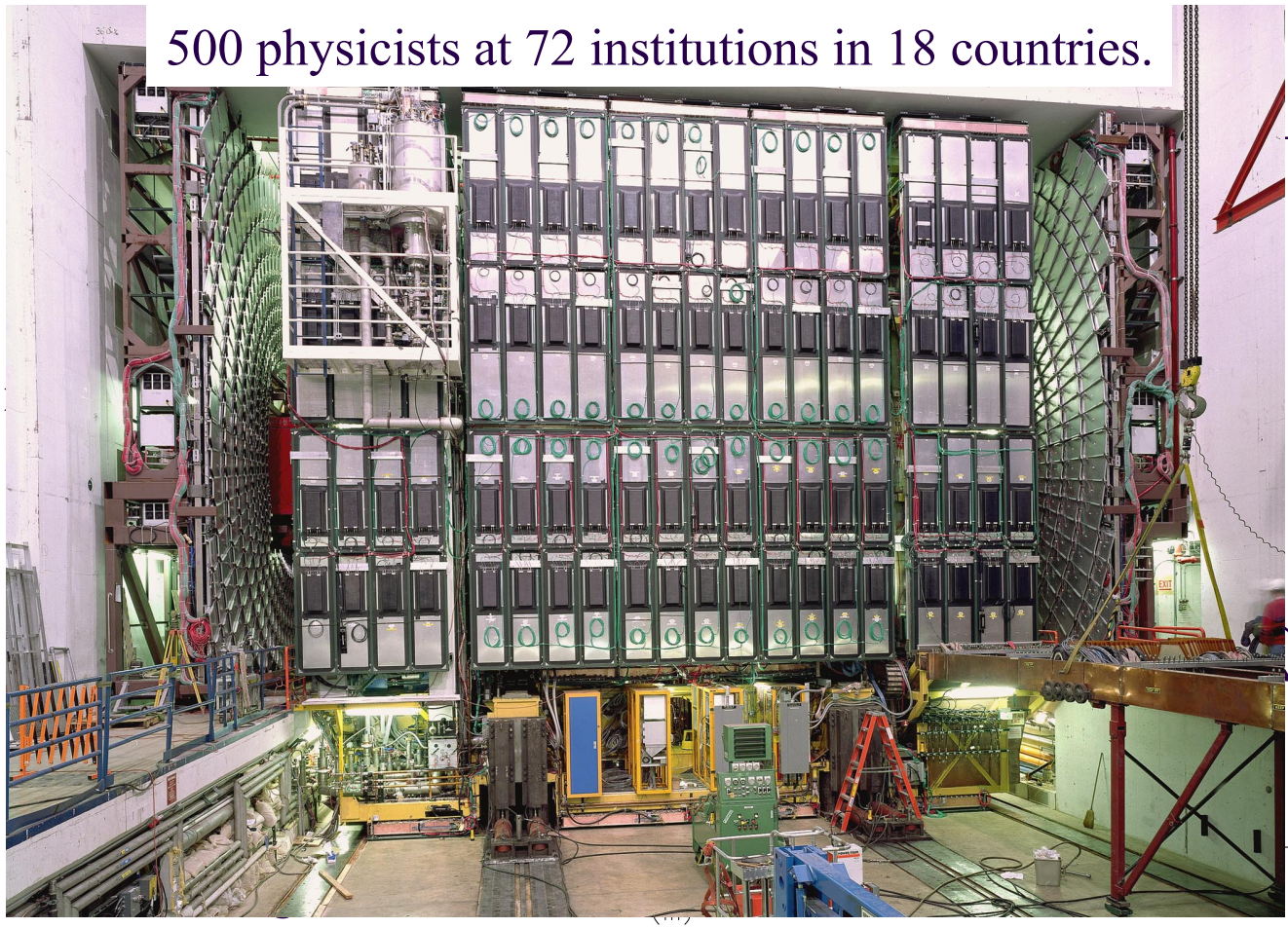


- Silicon Vertex Detector + Intermediate Silicon Layers
  - improved b-tagging capability out to $|\eta| = 2.0$
- Central Outer Tracker
  - open cell drift chamber, 44 cm < r < 132 cm, $|\eta| < 1.0$
- Upgrade Plug Calorimeter
  - replace gas calorimeters for $1.0<|\eta|< 3.0$,
- Muon Detectors
  - chambers added to close gaps in azimuthal coverage
- Electronics and Trigger, Data Acquisition
  - increase data transfer rates
  - level 3 processor farm analyses (and rejects on) full events
- Offline reconstruction and analysis software is completely rewritten in C++

525 physicists from 52 institutions from 11 countries

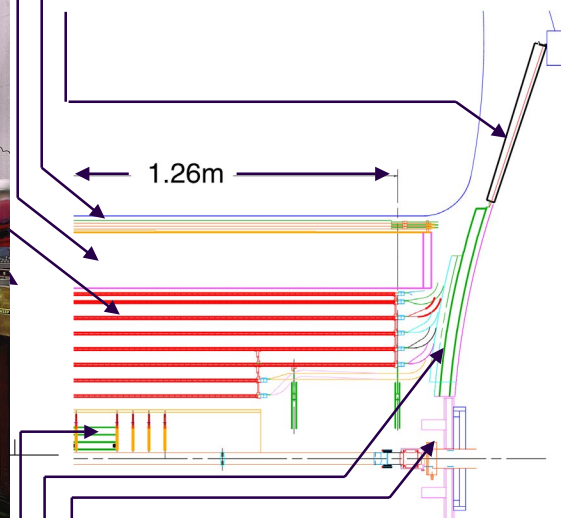
The upgraded DO Detector

500 physicists at 72 institutions in 18 countries.



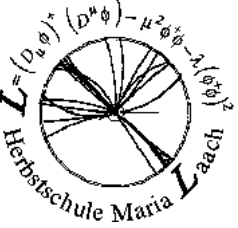
New Systems

- Fiber Tracker
- 2T Super conducting Solenoid
- Central Preshower
- Intercryostat Detector

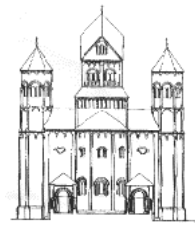


- Luminosity Monitor
- Forward Preshower
- Silicon Tracker

- Central Muon Scintillators
- Forward Muon Tracking
- Forward Muon Trigger Pixels
- 3 Level Trigger & DAQ
- Online System
- Offline Computing

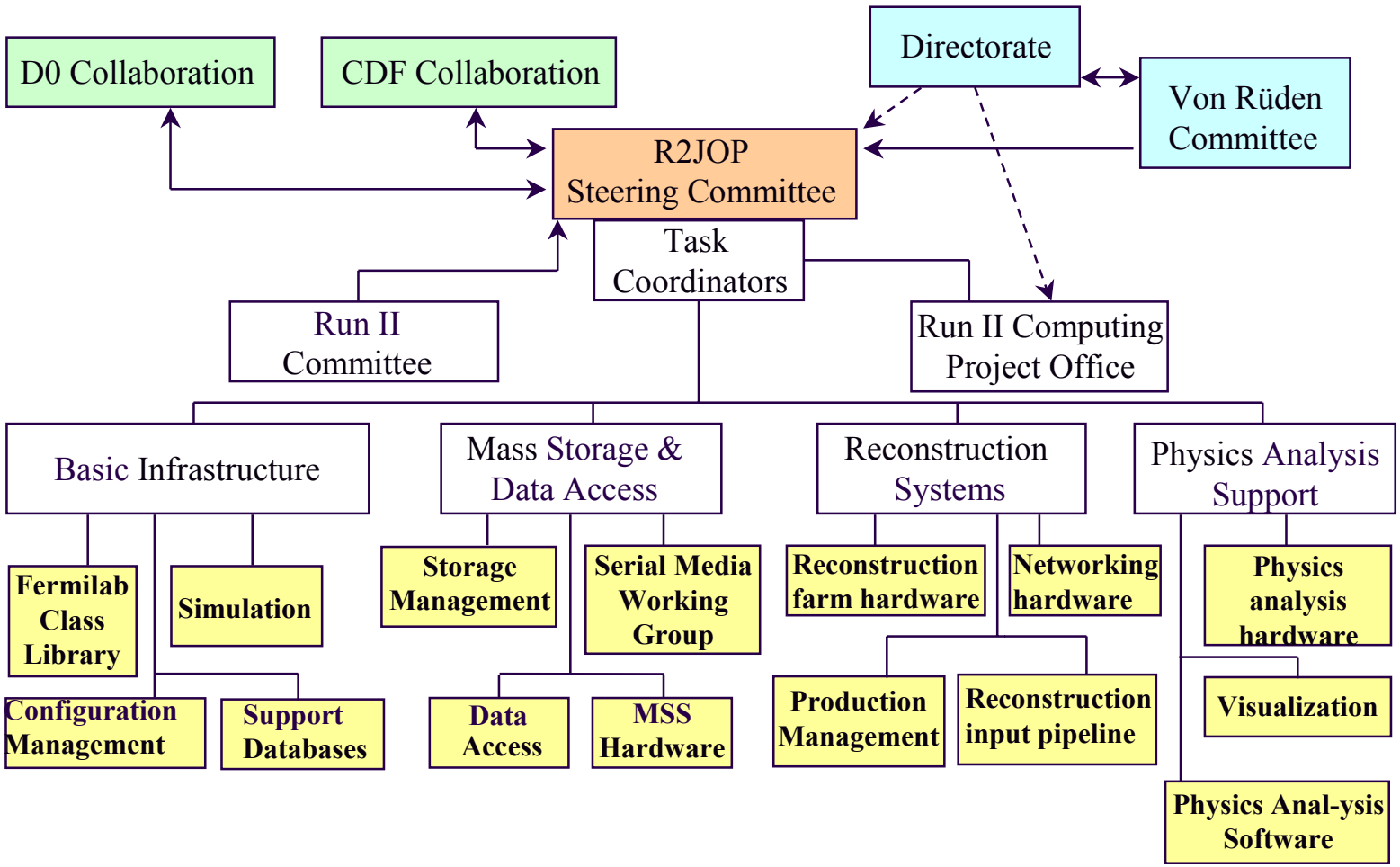


Computing Model for Run 2a



- CDF and D0 have similar but not identical computing models.
  - In both cases data is logged to tape stored in large robotic libraries.
  - Event reconstruction is performed on large Linux PC farms.
  - Analysis is performed on medium to large multi-processor computers
  - Final analysis, paper preparation, etc. is performed on Linux desktops or Windows desktops.

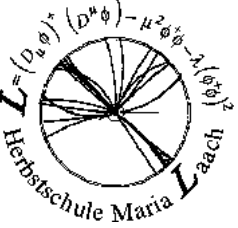
CDF/DO/CD: Run 2 Joint Project Organization



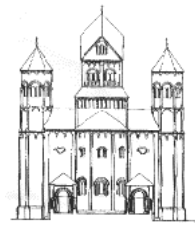
Run 2 Data Volumes

DAQ rates	Peak rate	53 Hz	75 Hz
	Avg. evt. Size	250 KB	250 KB
	Level 2 output	1000 Hz	300 Hz
	maximum log rate	Scalable	80 MB/s
Data storage	# of events	600M/year	900 M/year
	RAW data	150 TB/year	250 TB/year
	Reconstructed data tier	75 TB/year	135 TB/year

- First Run 2b costs estimates based on scaling arguments
  - Use predicted luminosity profile
  - Assume technology advance (Moore's law)
  - CPU and data storage requirements both scale with data volume stored
- Data volume depends on physics selection in trigger
  - Can vary between 1 – 8 PB (Run 2a: 1 PB) per experiment
- Have to start preparation by 2002/2003



Personnel for DØ computing (from DO talk)



The People We Have:

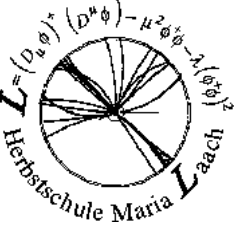
120 total

24 FNAL (DØ)	65 Senior
5 FNAL-CD	23 Postdocs
1 FNAL-BD	16 Students
90 Non-FNAL	16 CPs

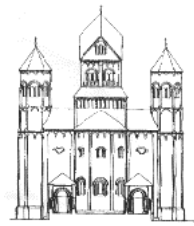
The People in the Joint Projects:

2.5 FTE	ZOOM (C++ libraries)
4.7 FTE	ENSTORE (Storage Management)
2.6 FTE	SAM (Data Access)
1.0 FTE	Configuration Management
1.5 FTE	Support Databases

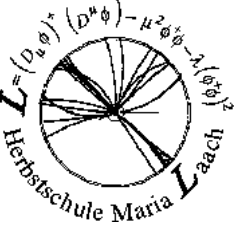
Other JP's -- Not closely tracking manpower, not as continuous (at this stage)



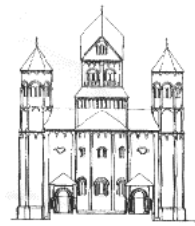
CDF/DO: C++ Experience



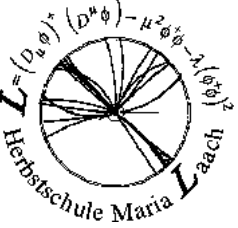
- Big change from procedural to object-oriented language.
- Some resistance.
- Large training requirements.
- Need for C++ experts to support the physicists on design and coding.
  - Two individuals were hired by Fermilab to provide that support.
- The code runs, is probably as fast or faster than Fortran code, and in general the exercise has been successful.
- Most (not all) new experiments choose C++ for offline event reconstruction.



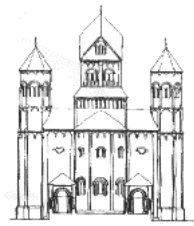
DO view: Successes, Worries, Regrets (from DO talk)



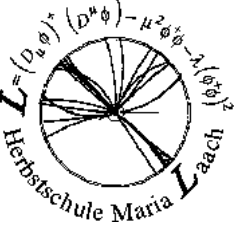
- The organization has worked well and proved flexible
- The decisions have been stable
- Are the decisions right?
- The Joint Offline Project has brought DØ and the CD into a much closer partnership
  - ZOOM, SAM, ENSTORE are good for the Lab as well as DØ
  - Expertise has been generally well targeted
- The Joint Offline Project is a lot of effort
  - Do we have the right balance between JOP effort and DØ effort?
- Major thing we'd change if we could:
  - More effort to support quality control from the Config Man end
  - More effort in software reviews



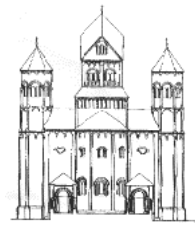
CDF/DO Experiences : Education



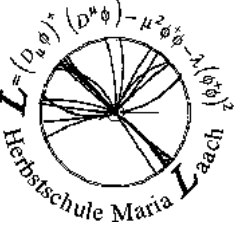
- Fermilab CD arranged C++ and OOAD classes from well-qualified Computer Science instructors
- Early differences
  - DØ emphasis on formal classes
  - CDF emphasis on good references, web communication
- Both may have converged to usual state of user-to-user transference?
- Bottom line, though, is that both experiments have retrained a substantial community, but not by any means all of their Run II users
- Doing better would be a big effort
  - Both experiments are always resource-limited when it comes to people; training and communication projects tend to be at the end of the line after the very early period



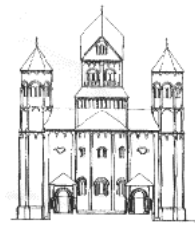
CDF/DO Experiences : Development Environments & Tools



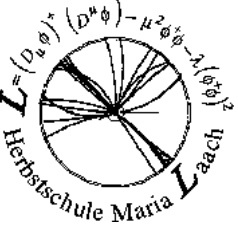
- Quest for a standards-compliant compiler
  - The state of C++ compilers in 97-98 was a BIG problem
  - We both chose the KAI compiler, with a much better approximation to standards compliance than native compilers or gcc (fortunately, it was available early on for Linux)
- Bringing in third party products
  - Open Inventor for KAI commissioned by Run II project
- Debugging complicates the issue -- not a good experience
- How many platforms is too many?
  - Run II has 2 offline platforms (IRIX and Linux) and 1 compiler for both platforms -- different SWITCH combinations alone mean that 2 $\rightarrow$ 6-20 different ZOOM & ROOT libraries are built (and tested) for Run II
  - * DØ uses NT for its Level 3 platform: an additional complication for the release system (recently changed to Linux)



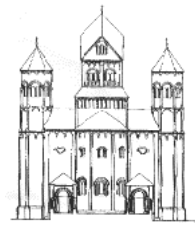
CDF/DO Experiences: Language and Design



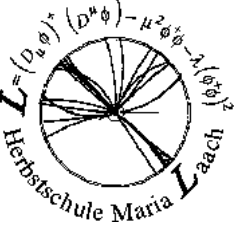
- Physical design
  - Importance very clear for making working releases all along the way
- General C++ design
  - Have an expert look over the design *before* starting to write:
    Plea from our OO experts to get first crack!
  - Portability is an issue with good and bad sides
    From a ZOOM developer: porting to ONE different compiler finds enough code problems to be well worth the effort
  - Design and code reviews become a must
    For the most part, reviews have been welcomed by developers
  - Memory management is *very* difficult for ex-Fortran programmers to master (current reconstruction still very sloppy)
  - From a L3 filter meeting: " I'm coding the xxx; it's going much more quickly than I thought, thanks to the beauty of C++ which lets me reuse all the code from yyy."



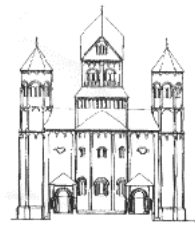
CDF/DO: Are There Lessons to Be Learned?



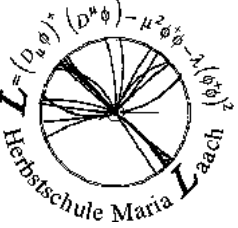
- Commonality of needs for infrastructure vs. divergence of tastes, interests, timescales:
  - not everything that could be done in common will be, but effort saved in a few areas is still worthwhile
- Common choice of compiler and release system enable joint work
  - development of RCP, e.g.
- Make infrastructure first
  - do it early to enable development but don't rule out redesign
- Pay attention to physical design
- Develop mechanisms for both little changes and big changes
  - if you plan for big changes, they are NOT too disabling to be contemplated
  - release strategy plays a big part



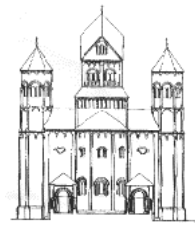
CDF/DO: Are We There Yet?
(from talk in 2000)



- Yes, we have successfully built large C++ systems
  - CDF: 1.3 million lines of code
  - DØ: 285 cvs packages
- Will the larger community find them highly usable or barely usable?
- Yes, we have build data handling systems that approach LHC sizes
  - 0.75 - 1.0 PB storage capacity (per exp't) will be available
  - Data movements of > 1 TB/day demonstrated with ENSTORE
  - DØ farm has seen 15 MB/sec data flow
  - CDF has exercised full online-offline chain, L3 to reconstruction
- Yes, we are keeping attention on integration and operation
  - ….and this is already paying off! A remark I hear frequently from members of both experiments: "I'm glad we are finding this out now and not a year from now!"



Other software for Run 2



- Mixture of commercial, lab-developed and open source.
- Each product is chosen based on its ability to solve a problem and on its cost (both to write and to support).
- Long list of products, some examples:
  - Linux, gcc, emacs, MySQL
  - KAI C++ compiler, LSF (Batch system), Purify
  - FBS, Enstore, SAM, ftt, ZOOM
  - GEANT3/4, ROOT

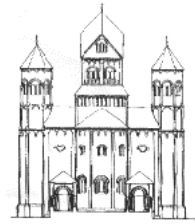
- End of day 1: 60 minutes

Computing** for High Energy Physics

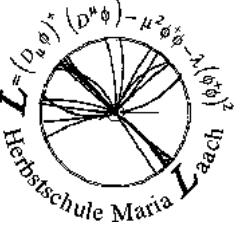
**Computing == Computing and Analysis

Part 2

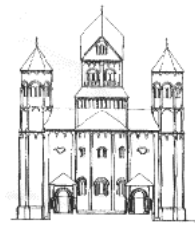
Past and Present Strategies for data processing/data handling



- Use 'Commodity' components where possible
  - inexpensive CPUs in 'farms' for reconstruction processing e.g. PCs
  - inexpensive (if somewhat unreliable) tape drives and media
- Multi-vendor
  - IBM, SGI, DEC, SUN, Intel PCs
- Use much Open Source Software (Linux,GNU, tcl/tk, python, apache,CORBA implementations…)
- Hierarchy of active data stores
  - Disk, Tape in Robot, Tape on Shelf
- Careful placement and categorization of data on physical medium
  - optimize for future access patterns



Choices for CPU resources



SMP computers

Commodity computers / cluster

Situation is changing quickly with time

- For commodity systems with Linux:
  - Improved system admin software becomes available
  - Improved development tools become available
  - Vendors start to offer integrated solutions (not always for our problems…)
  - Better resource scheduling software available
- Analysis code is slower than expected (less I/O bound)
  - May become effective to access data via network (cross-mounting, global file systems, …)
- Disk is getting cheaper:
  - May be affordable to store multiple copies of data

Things taken for granted (I): hardware

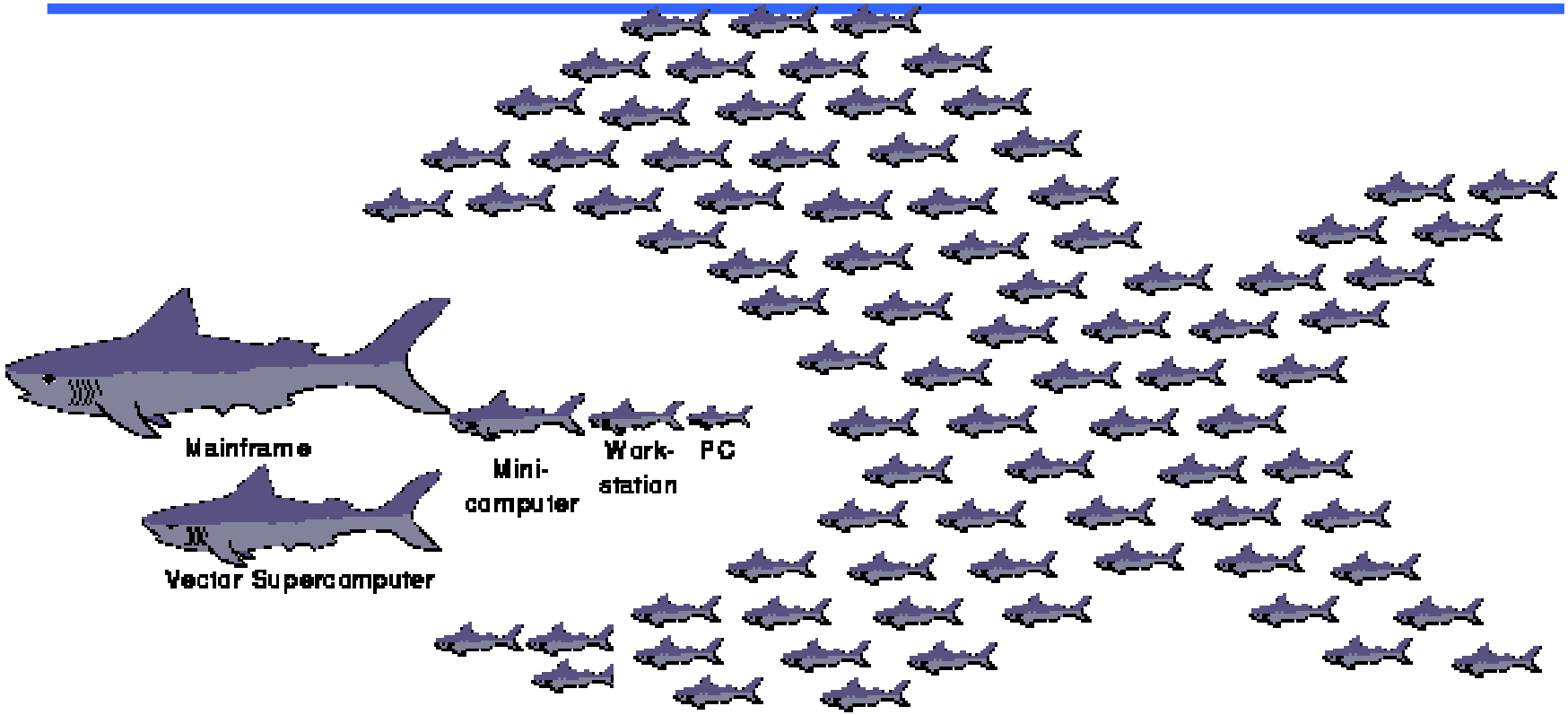
- PC+Linux: the (easily assembled) new supercomputer for scientific applications

obswww.unige.ch/~pfennige/gravitor/gravitor_e.html

www.cs.sandia.gov/cplant/

Paris Sphicas
CHEP2000

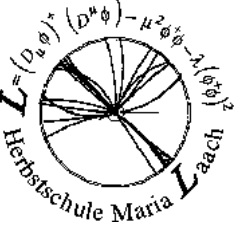
Hardware II: The new Supercomputer



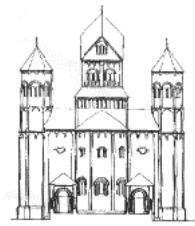
NOW

Found at the NOW project (http://now.cs.berkeley.edu)

Paris Sphicas
CHEP2000

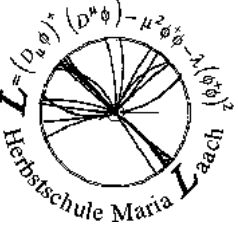


Hardware (aka "enough CPU")

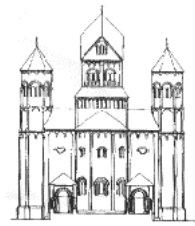


- Explosion of number of farms installed
  - Very cost-effective
    - Linux is free; PC's are inexpensive
    - Interconnect: Fast/Giga Ethernet, Myrinet, Fibrechannel, even ATM
  - Despite recent growth, it's a mature process
    - Basic elements (PC, Linux, Network) are all mature technologies.
      - Problem solved.
    - But still left: Control & Monitor of thousands of (intelligent) things
    - But C&M does not seem to be a fundamental problem
- Conclusion on hardware: probably rightly skipped
  - It's the software that's harder to design, code and operate
    - And anyway the industry is many times better than us

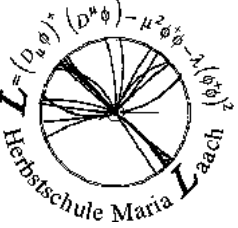
Paris Sphicas
CHEP2000



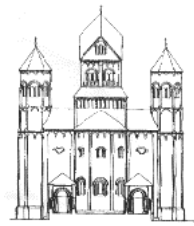
Key Elements of Run II Data Access



- How do we work with the data?
- How do we physically organize the data?
- On what do we store it - where?
- How do we migrate between parts of the storage hierarchy ?
- How do we provide intelligent and controlled access for large numbers of scientists?
- How do we work with the data ?



Access to Objects - OO design



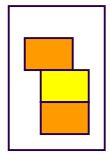
- C++ Reconstruction and Analysis Programs
  - Fully object oriented design - STL, Templates (D0)
  - reference-counted pointers (D0)
  - OO data model
  - like OODBMS persistent objects inherit from persistent class
- Objects and Collections of Objects stored persistently to disk and tape
  - 'flattened' out to files in special HEP formats
- d0om persistency package for D0
  - supports various external 'flattened' format, including relational database
  - allows for possibility of storing some 'tiers' of the data in OO database if proven useful
- ROOT (HEP analysis package) file format for CDF
- Schema evolution can be tailored to need

Data Tiers for a single Event (DO)

~200B

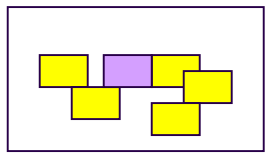
Data Catalog entry

5-15KB



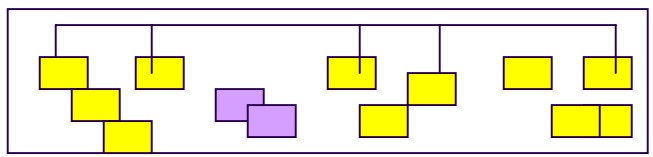
Condensed summary physics data

50-100KB



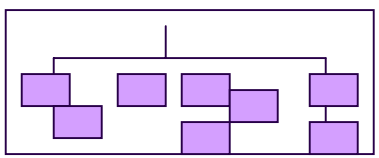
Summary Physics Objects

~350KB



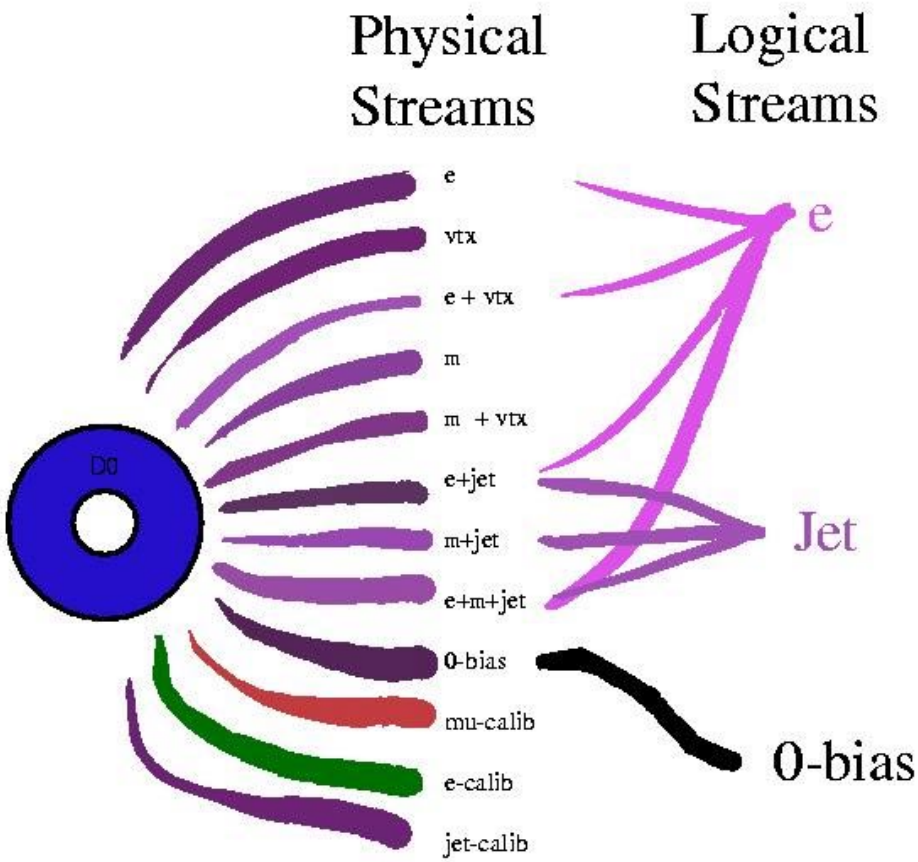
Reconstructed Data -
Hits, Tracks, Clusters,Particles

250KB



RAW detector measurements

Streaming the Data - optimize for data access traversal (D0)



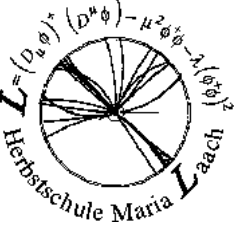
Events with more than one trigger classification go to overlap streams

D0 approach to streaming the data

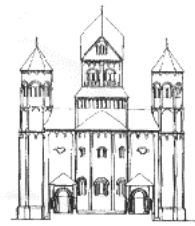
Up-front physical data organization and clustering

Multiple streams written and read in parallel

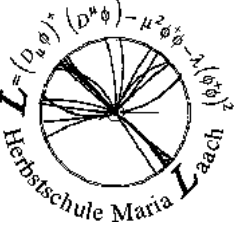
Streams are physics based, unlike disk striping



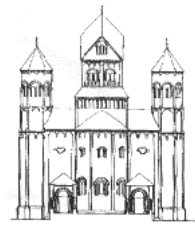
Example File and Event Catalog for Run II (DO)



- Oracle 8 database ==> 0.5 - 1 TB for each experiment, including detector run conditions and calibration data
  - 1.8 10**9 Event metadata entries, bit indexes, own data types
  - several million file entries
- Oracle Network site-wide license - now on Linux too
- Hot backups with Recovery Manager
- Data Files Catalogued and related to
  - Runs and Run conditions
  - Luminosity information about the accelerator
  - The processes which produced (and consumed the data)
  - Detector geometry, alignment and calibration data

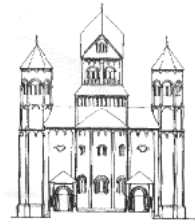


Run II Data Access - strategies

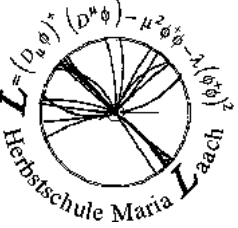


- data content for an event from different processing stages stored in different physical collections
  - 'tiers' of data of different sizes and content - RAW, fully reconstructed, summary reconstructed, highly condensed summary, ntuples and meta-data
- primarily file-oriented access mechanisms
  - fetch a whole collection of event data (i.e. 1 file ~ 1GB)
  - read through and process it sequentially
- optimize traversal of data & control access based on physics & user - not on file system
- use relational databases (Oracle centrally) for file and event catalogs and other 'detector conditions' and calibration data (0.5 - 1 TB)
- import simulated data (files and tapes) from MC

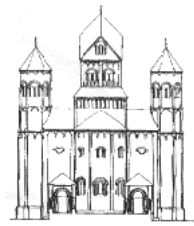
D0 Fully Distributed Network-centric Data Handling System



- D0 designed a distributed system from the outset
- D0 took a different/orthogonal approach to CDF
  - Network-attached tapes (via a Mass Storage System)
  - Locally accessible disk caches
- The data handling system is working and installed at 13 different 'Stations' – 6 at Fermilab, 5 in Europe and 2 in US (plus several test installations)



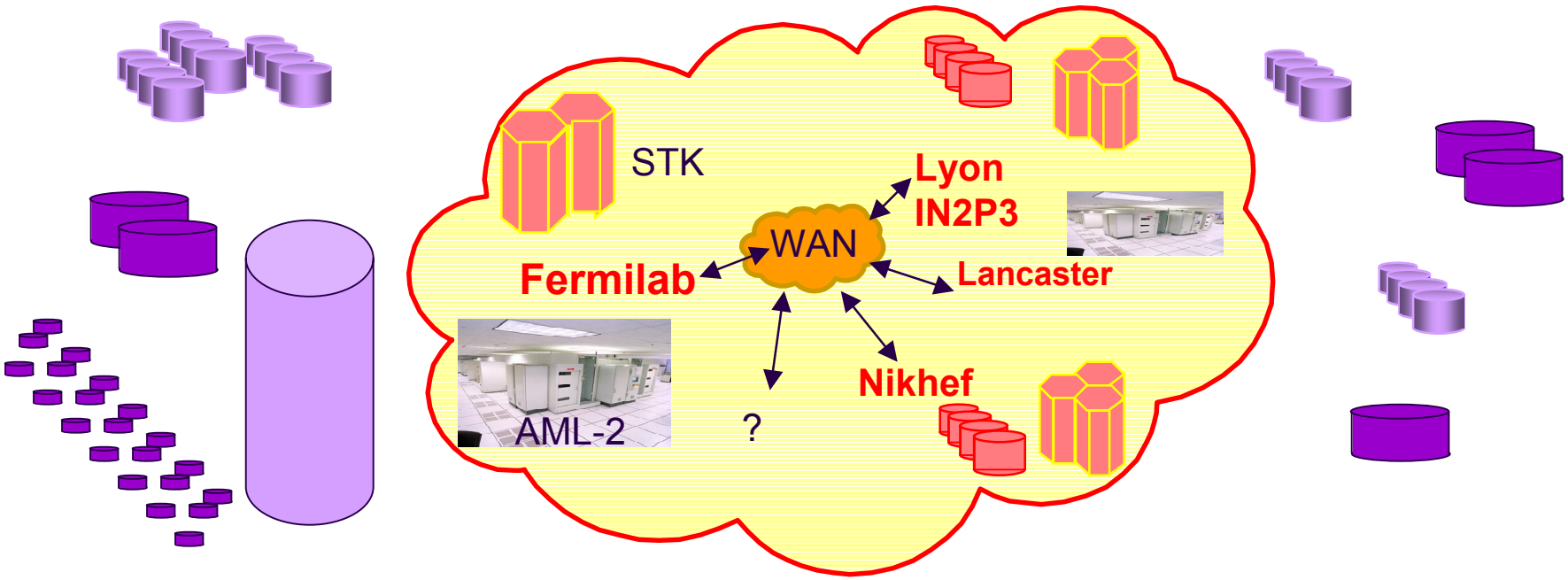
A DO-specific Data Grid System



- Many CPUs (from Origin2000s to Compaq Alphas, Farms of Linux boxes, Linux Analysis Clusters and desktops) + Network switches + LAN + WAN all integrated together to make a working Data Grid System
  - Data Grid Architecture
- 360 registered users using 212 registered nodes(~500 cpus), have defined 6600+ datasets and run 34,000+ jobs to look at >35TB of stored data (~8 TB from detector data)

The Data Store and Disk Caches

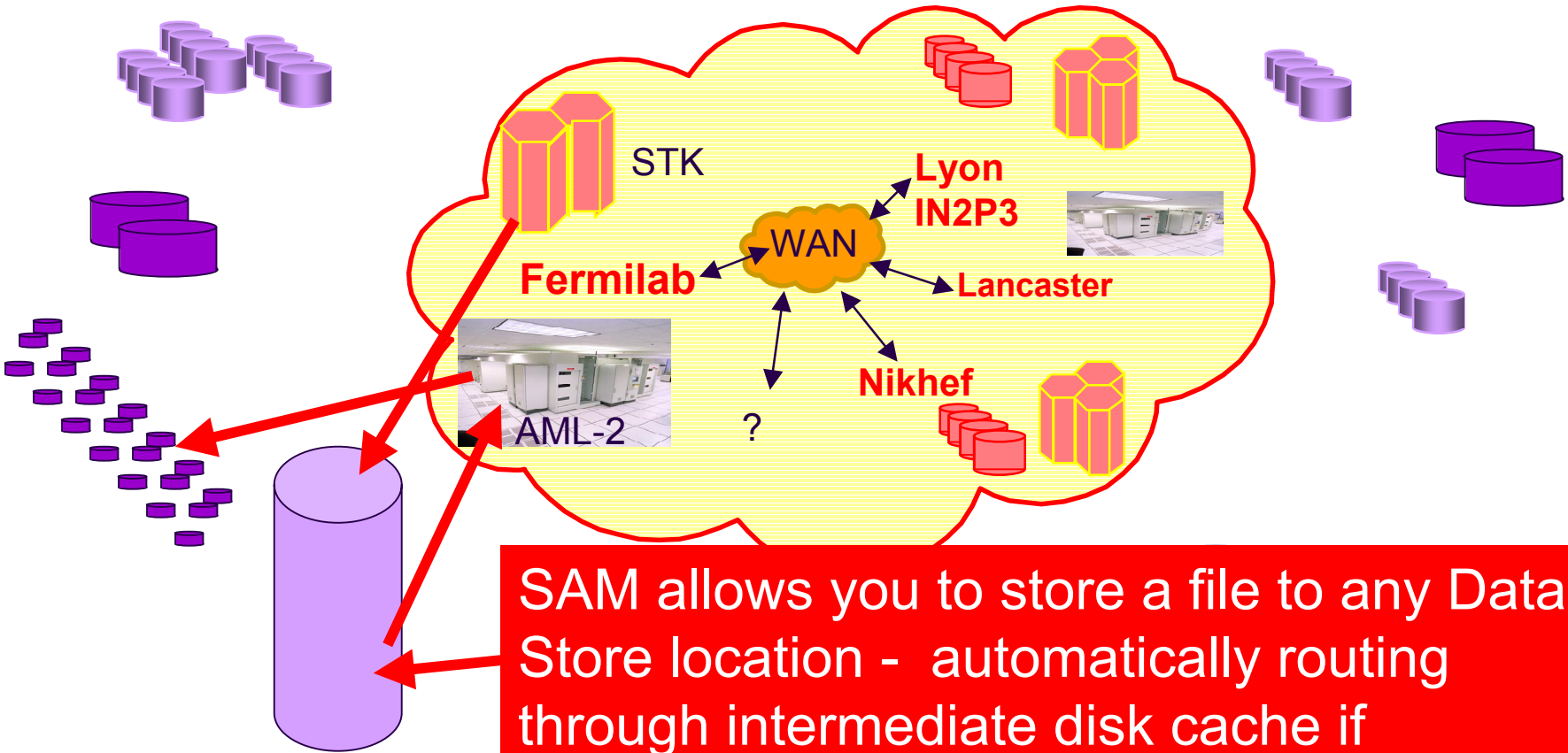
Data Store stores read-only Files on permanent tape or disk storage



All processing jobs read sequentially from locally attached disk cache.
"**S**equential **A**ccess through **M**etadata" – **SAM**
Input to all processing jobs is a Dataset

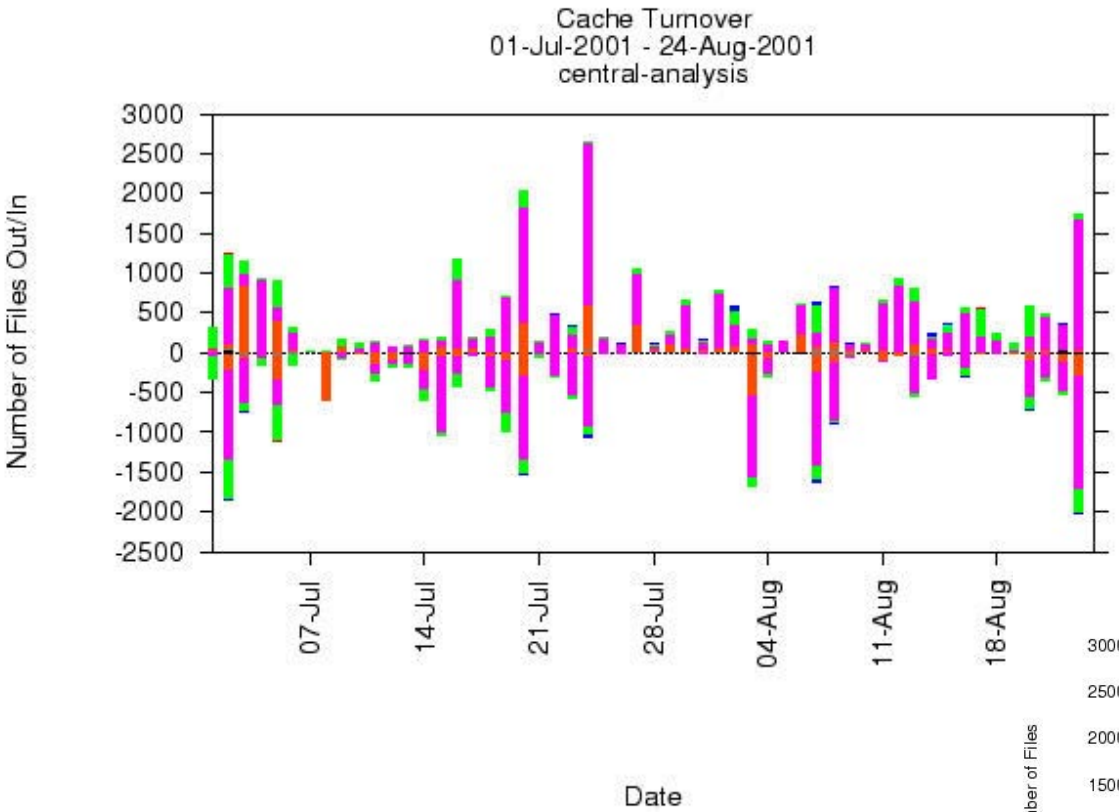
Event level access is built on top of file level access using catalog/index

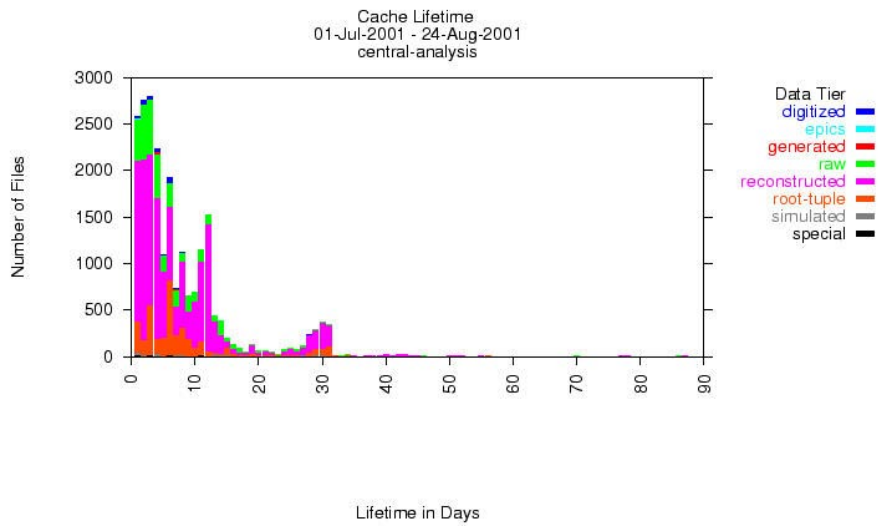
The Data Store and Disk Caches



SAM allows you to store a file to any Data Store location - automatically routing through intermediate disk cache if necessary and handling all errors/retries

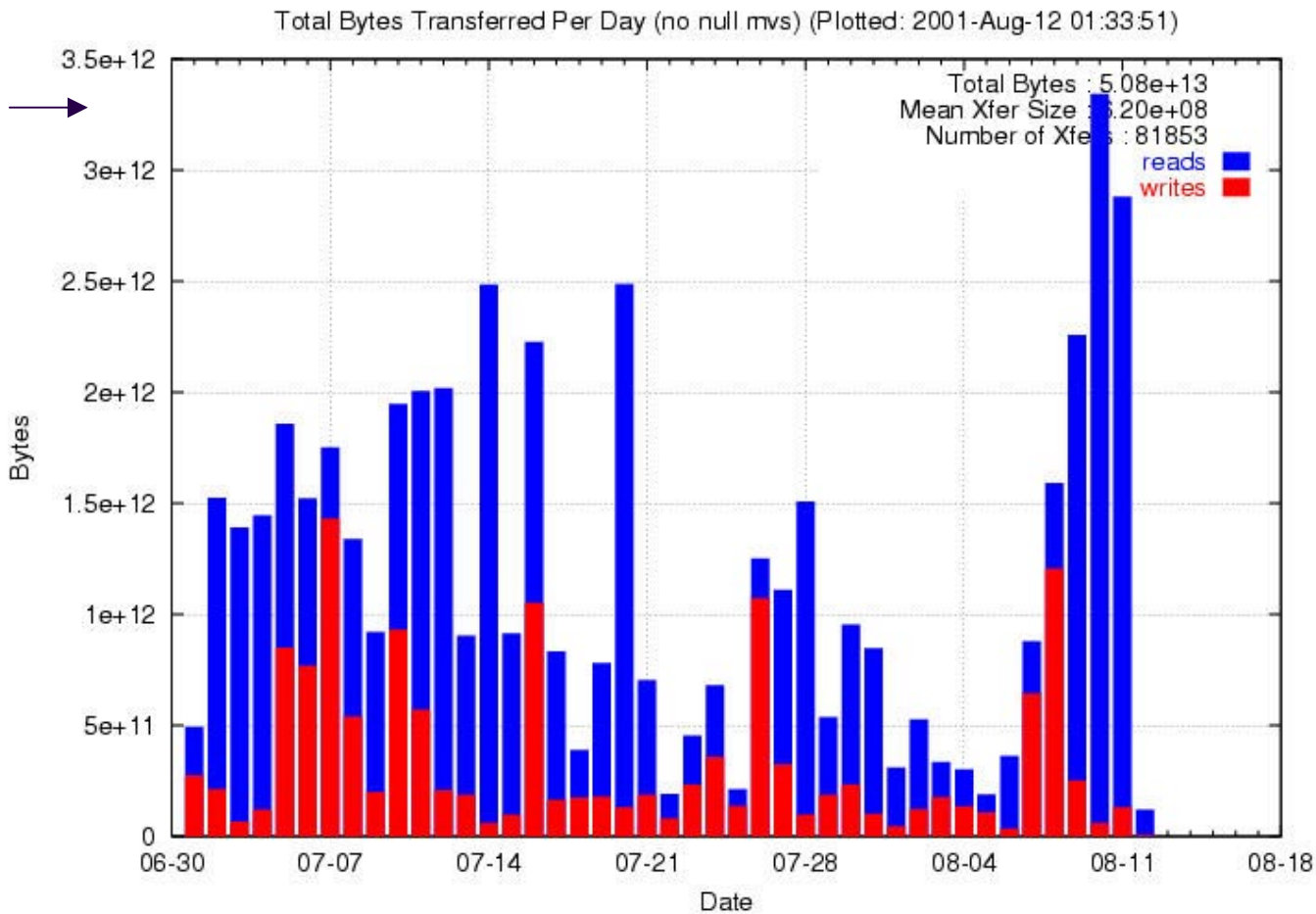
Central Analysis Cache turnover and lifetime



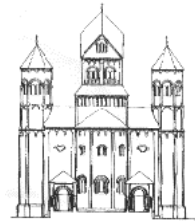


Enstore – recent Bytes in/out per day

3.4TB/Day →

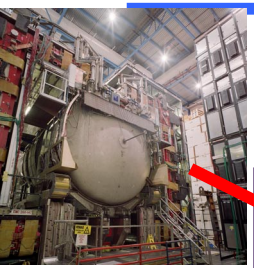


The Network is the Heart of the System



- Files are moved via LAN or WAN in the same manner – using various file transfer protocols over an IP packet network.
  - encp, bbftp (7 way parallel transfers), rcp, hpss form of cp, etc.
- Fermilab site sees greatest movement of data.
- Enstore file transfer protocol (encp) provides load balancing between multiple network interface cards
  - For Origin2000 each Gbit Ethernet interface needs 1 dedicated CPU and supports ~30MB/sec
- World-wide DØ Monte Carlo Production is up and working now
  - Current total Bandwidth to Fermilab ~50-100Mb/sec
  - Shipping MC data back and forth is essential
    - **Total Bandwidth ~200Mb/sec (2001)**
    - **Total Bandwidth ~400Mb/sec (2002)**
  - Real data processing at remote farms + reprocessing (?)
    - **Total Bandwidth ~800Mb/sec (2002)**
    - **Total Bandwidth ~(1200/1600/3200/4000)Mb/sec (2003/4/5/6)**
- Is Trans-Atlantic bandwidth available?

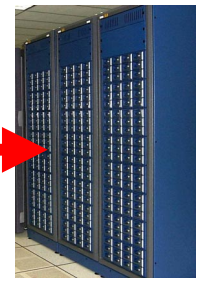
SAM Processing Stations at Fermilab



"data-logger"

"d0-test" and "sam-cluster"

"central-analysis"



12-20 MBps

100+ MBps

400+ MBps

12-20 MBps

"farm"

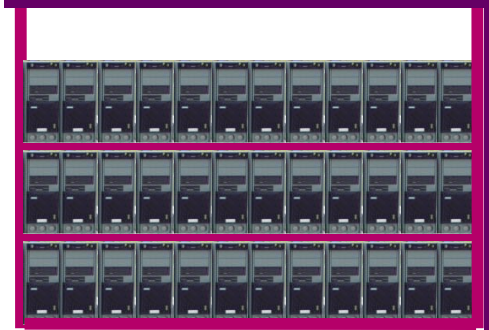
"linux-analysis -clusters"

"linux-build-cluster"

"clueD0" ~100 desktops

SAM Processing Stations at Fermilab

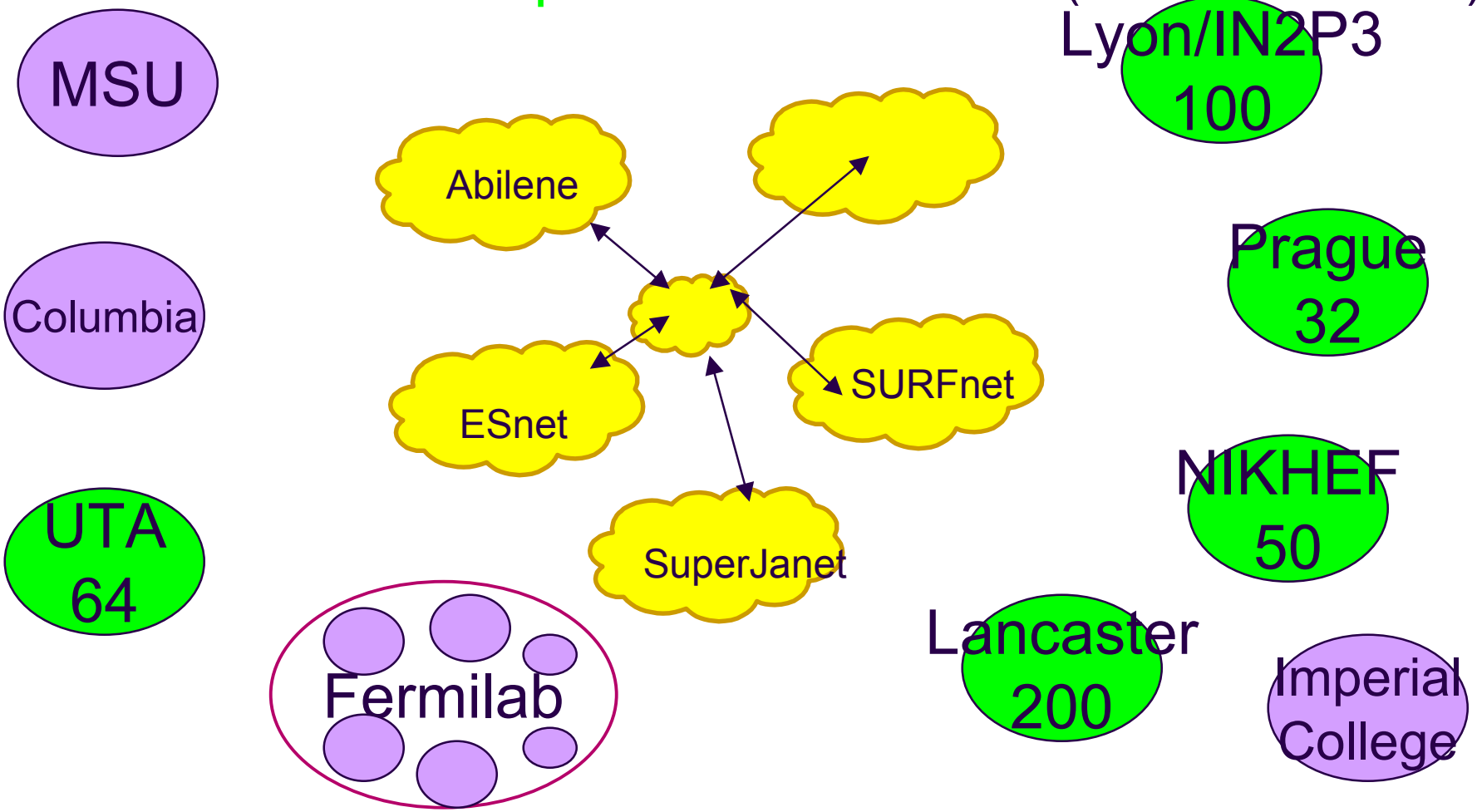
Origin2000 has for Online
176 300 MHz IP37 processors · GS80 with
5 Gbit Ethernets + 100Mbits ha EV67 CPUs
6 Fibrechannel controllers Mbit Ethernet
27 TB disk (5TB SAM Cache)
45 GB memory
User login – NFS mounted er



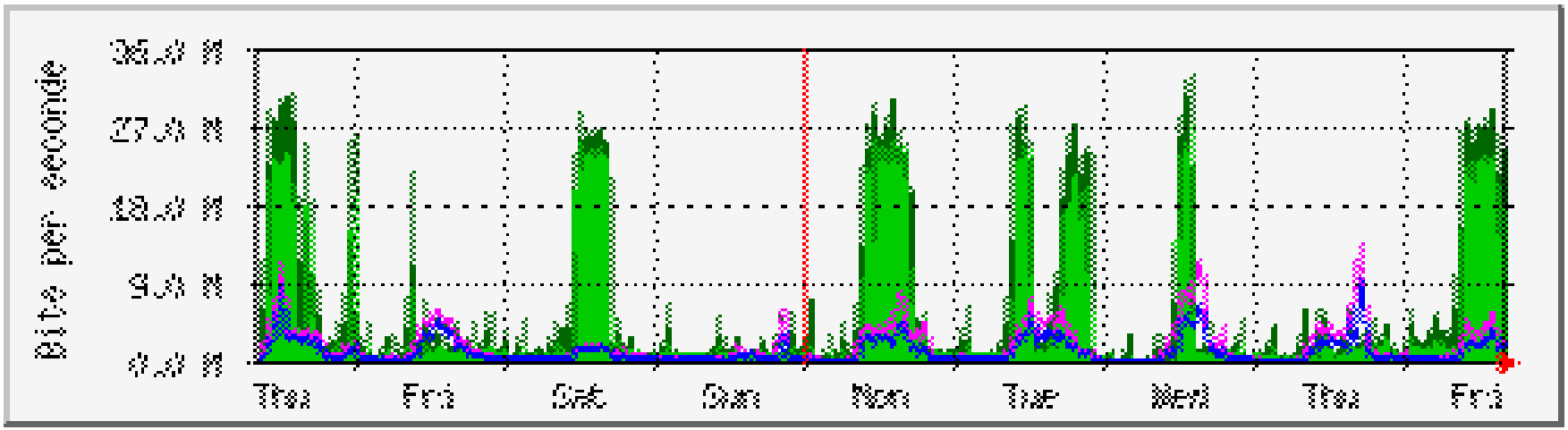
512MB/1GB memory
2 * 16 GB disks on each
SGI O2000 file server
FBS batch system

DO Processing Stations Worldwide

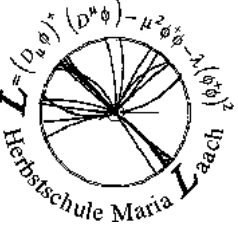
= MC production centers (#nodes all duals)



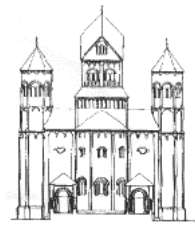
DO MC file transfers from NIKHEF to Fermilab



Green shows periodic file movement from NIKHEF to Fermilab, under SAM control, using a single bbftp session with 7-way parallel transfer. Week of July 20, 2001.

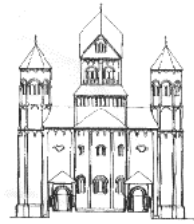


D0: Some Additional Features of SAM

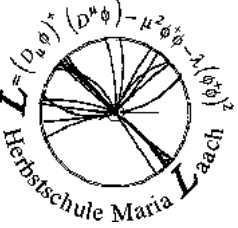


- Maintain detailed processing information – from the trigger through to individual and group datasets – almost enough for 'virtual data'. Very rich meta-data!
- Never directly access the database – only via CORBA server. CORBA interfaces between all components.
- Provide hooks and knobs for resource management
  - Prioritization of access modes
  - Fair sharing of various resources by group or activity
  - Interface to 'generic' batch system with specific implementations for LSF, FBS + PBS, Condor (under test)
  - Various access optimizations and controls
- Automate error recovery/restarts.
- Build for the future – architecture and components that can evolve to Grid standards.

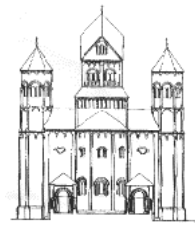
DO Experiences – Are we meeting the goals?



- Provide reliable and robust storage of the raw detector data, MC data and other derived data
  - Enstore works well – even with flakey tape drives (Mammoth 2)
  - In October will write raw data to STK9940
  - LTO for Monte Carlo data very soon.
- Keep up with production processing. Be able to process raw data files within minutes of writing them to tape
  - We have the capability to do this.
  - Can read raw data into disk cache on farms, or other Stations



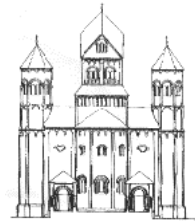
D0 Experiences – Are we meeting the goals?



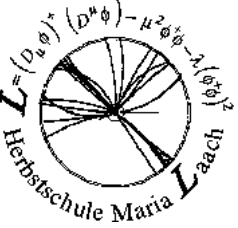
- Provide easy, rapid and intuitive access to data on a variety of systems at Fermilab and at remote locations where processing and data storage resources are available to D0 physicists
  - Almost – must transition from datasets of commissioning phase, with their complex structured queries that describe the data, …to physics analysis dataset specifications e.g.

    trigger JET_MULTI root-tuples with reco_version>=p04.01.00 and tuple_maker_version=p04.07.02
  - Exact same way of accessing data and storing back resultant files on every station worldwide. Transparent access to the files in a dataset (in random order) through the standard framework packages - just specify input file SAMInput:

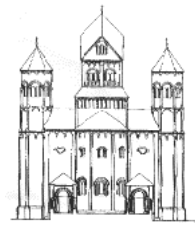
DO Experiences – Are we meeting the goals?



- Provide accurate detailed information on the processing steps that transformed event data – from the trigger through reconstruction and all the way to the creation of individual or group datasets
  - This is really getting there! Almost enough for “virtual data”
- Help enable and encourage worldwide participation in MC production, data processing and analysis
  - The “Station” strategy and architecture has encouraged and facilitated MC data production
  - We feel sure it will facilitate analysis of data worldwide, by a greater number of physicists

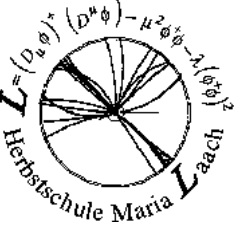


Future Plans and Conclusions



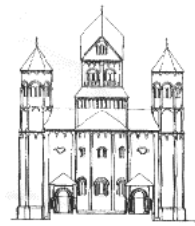
- Job management and submission must be enhanced => Condor +
- Need to more *intelligently manage* all the resources.
- User identification and security using Globus tools. Also GridFTP.
- Must make sure the whole system is fully scalable – potentially to all 72 institutions on D0 and very easy to install and configure
  - e.g Database caching and partial replication strategies
  - Enhance and distribute monitoring framework (Grid tools?)
- Event level access to data must be implemented.
- Integration with Root framework is almost done.

D0 can contribute to and also take advantage of Grid tools and projects (PPDG in particular). The D0 system forms an excellent real-world production testing ground as we evolve to a full collaboration-wide D0 Data Grid in the next 12 to 18 months.



Run IIa Equipment Spending Profile

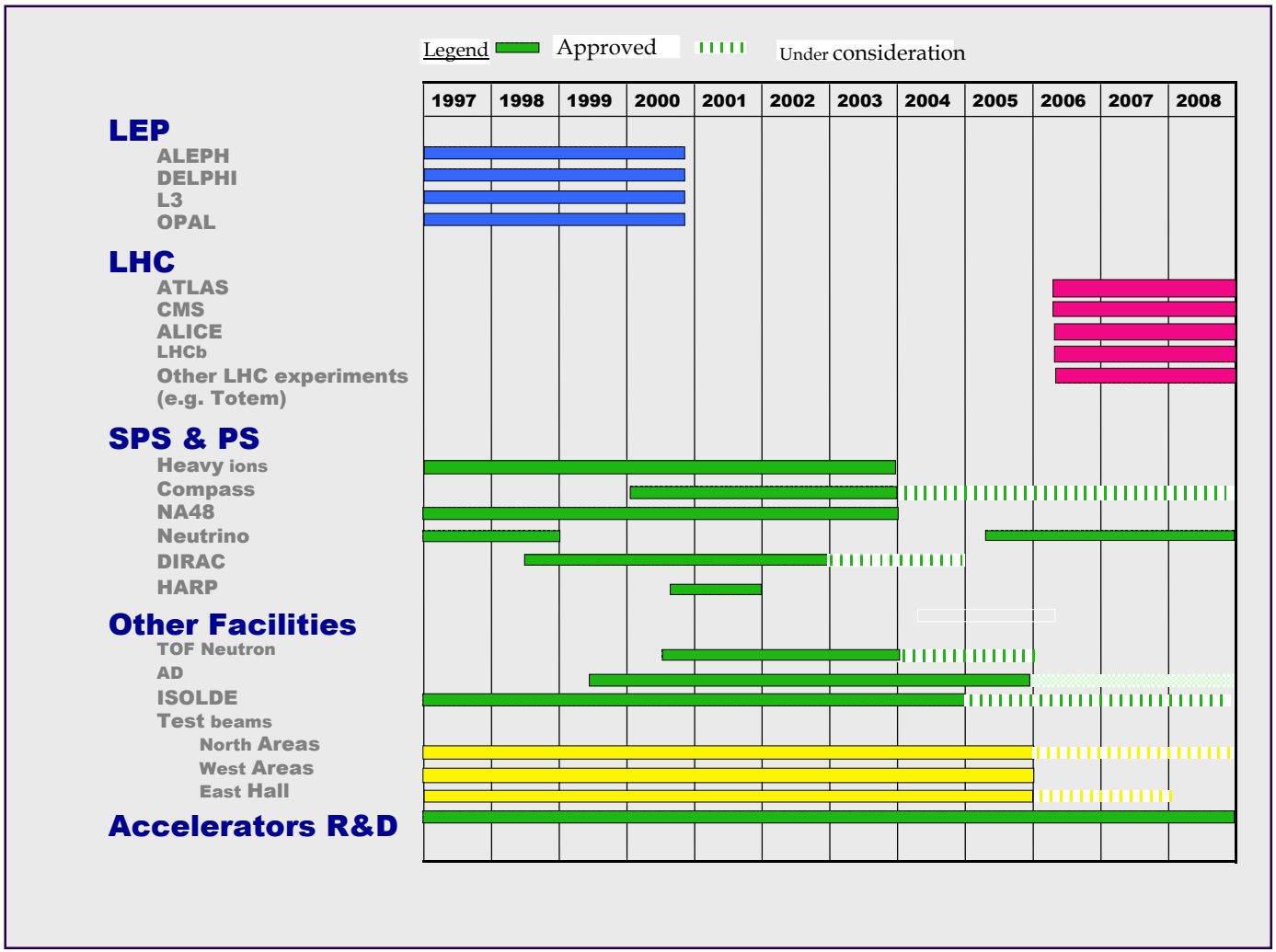
(Total for both CDF & D0 experiments)



- Mass storage: robotics, tape drives + interface computing.
- Production farms
- Analysis computers: support for many users for high statistics analysis (single system image, multi-CPU).
- Disk storage: permanent storage for frequently accessed data, staging pool for data stored on tape.
- Miscellaneous: networking, infrastructure, ...

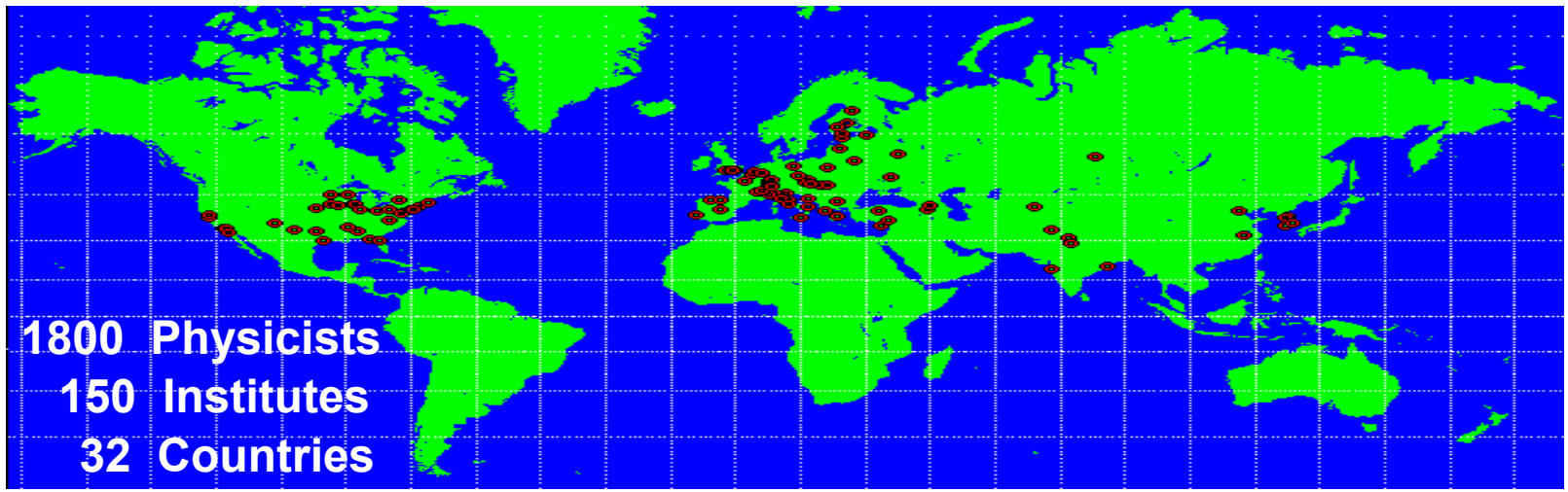
Fiscal Year	MSS	Farms	Analysis	Disk	Misc	Total (both)
Spent in FY98	$1.2M	$200K	-	$200K	$400K	$2M
Spent in FY99	$2.2M	$700K	$2M	$800K	$300K	$6M
Spent in FY00	$450K	$350K	$100K	$300K	$800K	$2M
Budget FY01	*$450K*	*$350K*	*$2.14M*	*$690K*	*$70K*	*$4M*
Plan for FY02	*$500K*	*$1.2M*	*$2.16M*	*$610K*	*$30K*	*$4.2M*
Total Needs	*$4.8M*	*$2.8M*	*$6.4M*	*$2.6M*	*$1.6M*	*$18.2M*
Continuing Operations (FY2002 and beyond)						*$2M*

The CERN Scientific Programme



CMS Computing Challenges

- Experiment in preparation at CERN/Switzerland
- Strong US participation: ~20%
- Startup: by 2005/2006, will run for 15+ years



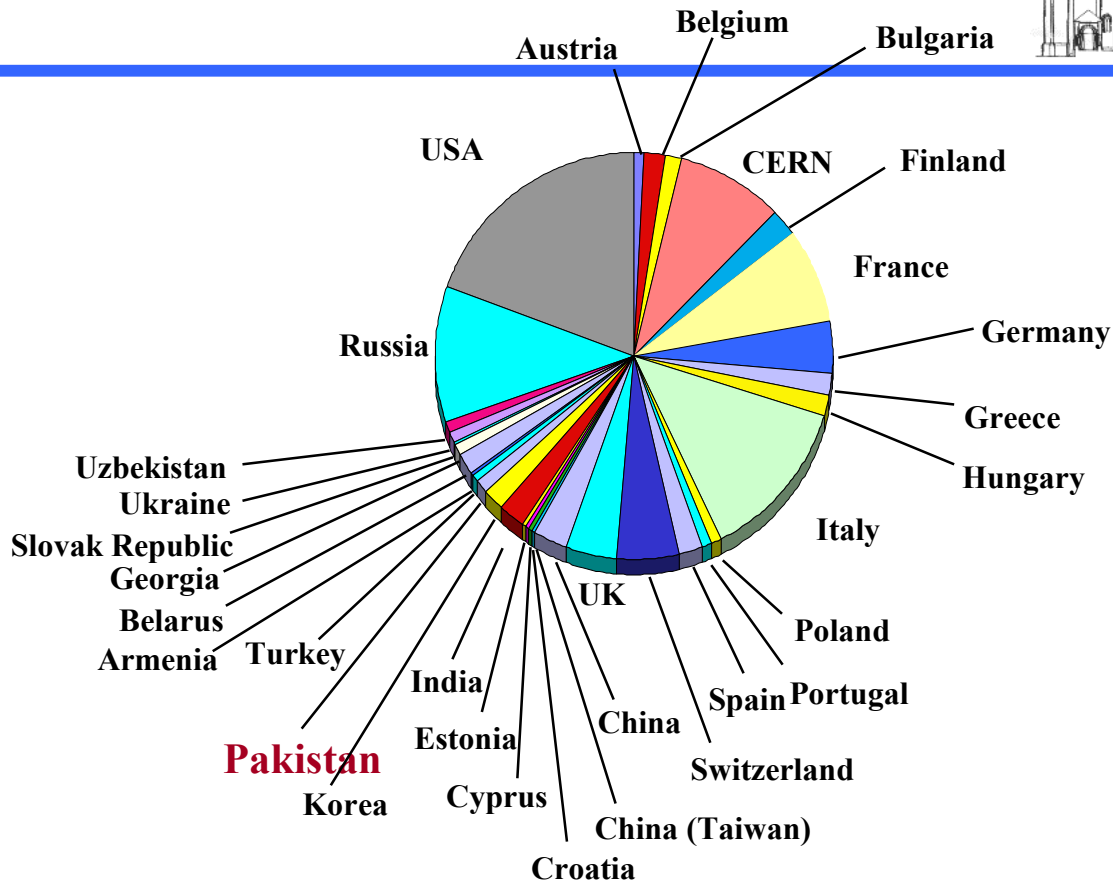
Major challenges associated with:
Communication and collaboration at a distance
Distributed computing resources
Remote software development and physics analysis
R&D: New Forms of Distributed Systems

The CMS Collaboration

	Number of Laboratories
Member States	58
Non-Member States	50
USA	36
Total	144

	Number of Scientists
Member States	1010
Non-Member States	448
USA	351
Total	1809

Associated Institutes	
Number of Scientists	36
Number of Laboratories	5

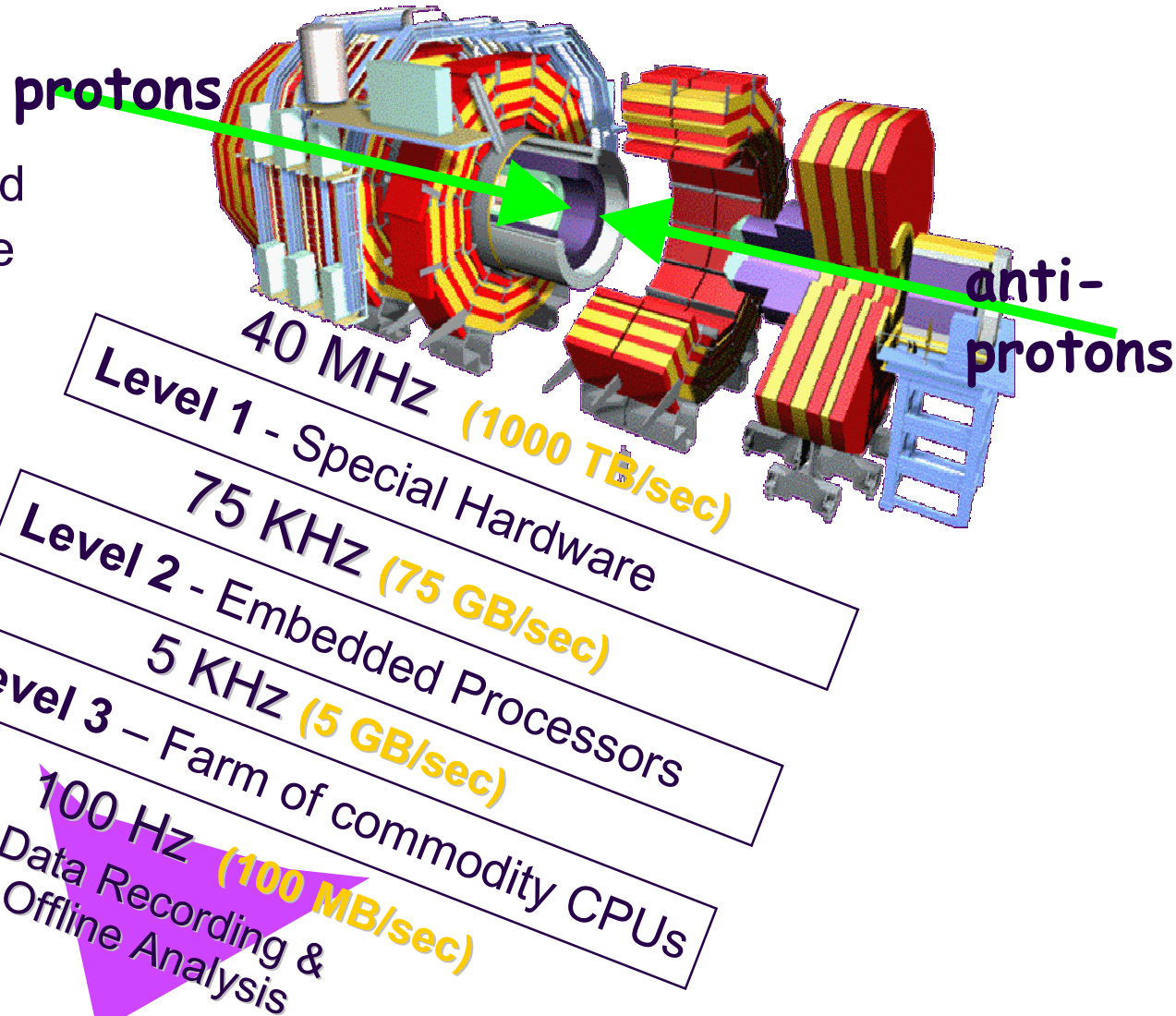


1809 Physicists and Engineers
31 Countries
144 Institutions

Data reduction and recording: here CMS in 2006

On-line System

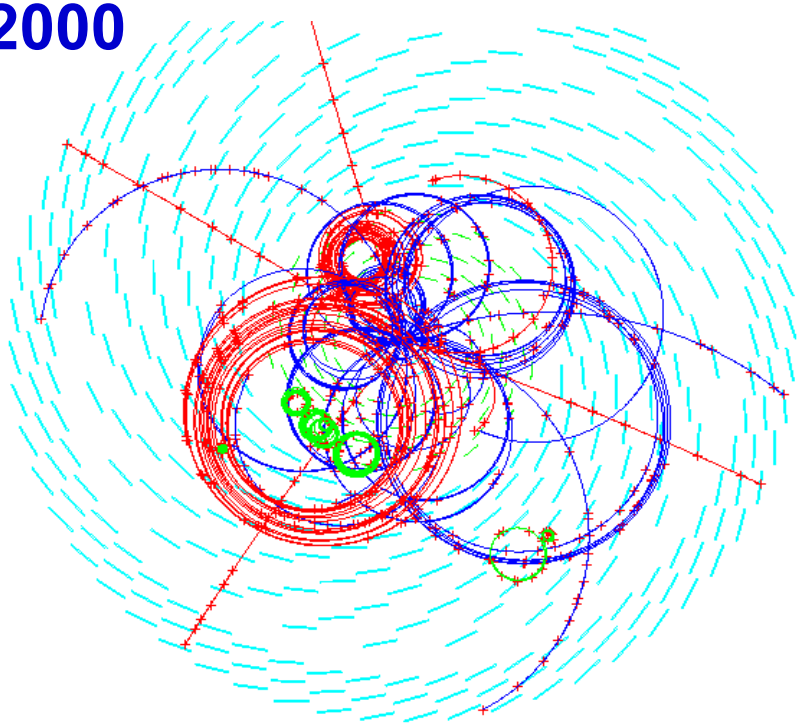
- Multi-level trigger
- Filter out background
- Reduce data volume
- 24 x 7 operation



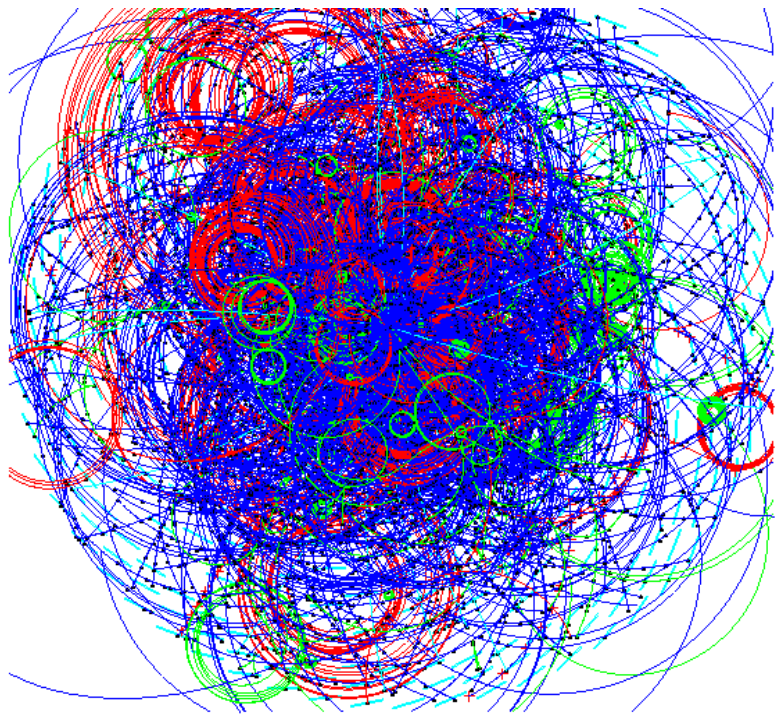
LHC Data Complexity

- "Events" resulting from beam-beam collisions:
  - Signal event is obscured by 20 overlapping uninteresting collisions in same crossing
  - CPU time does not scale from previous generations

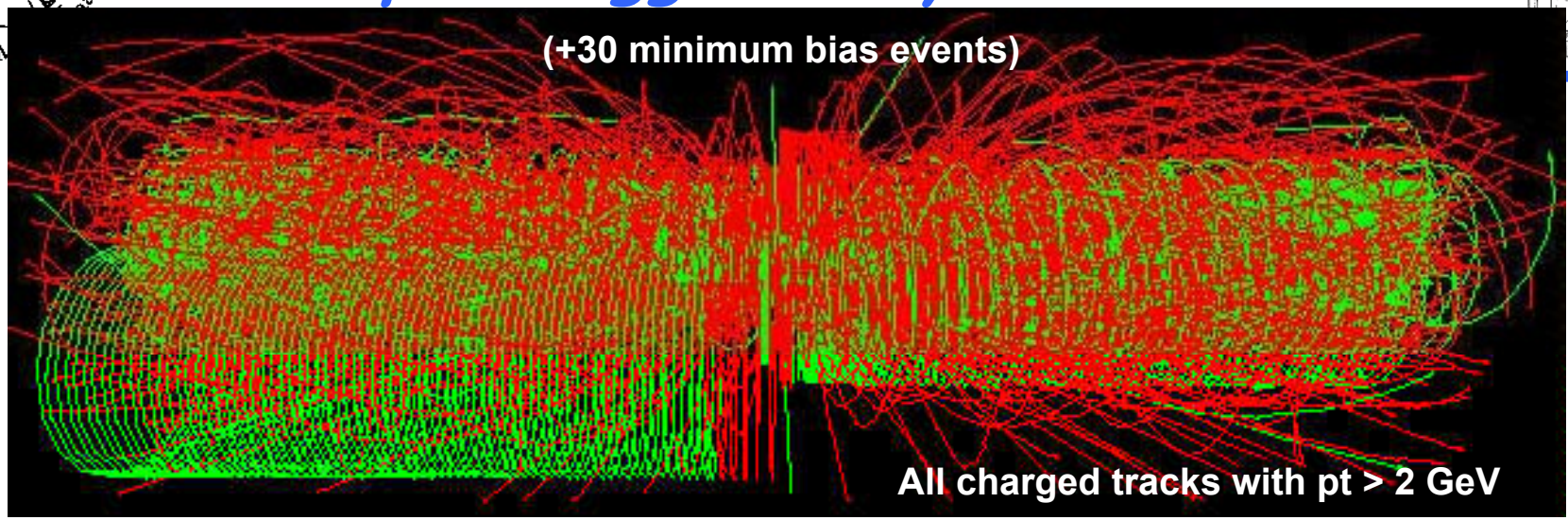
2000

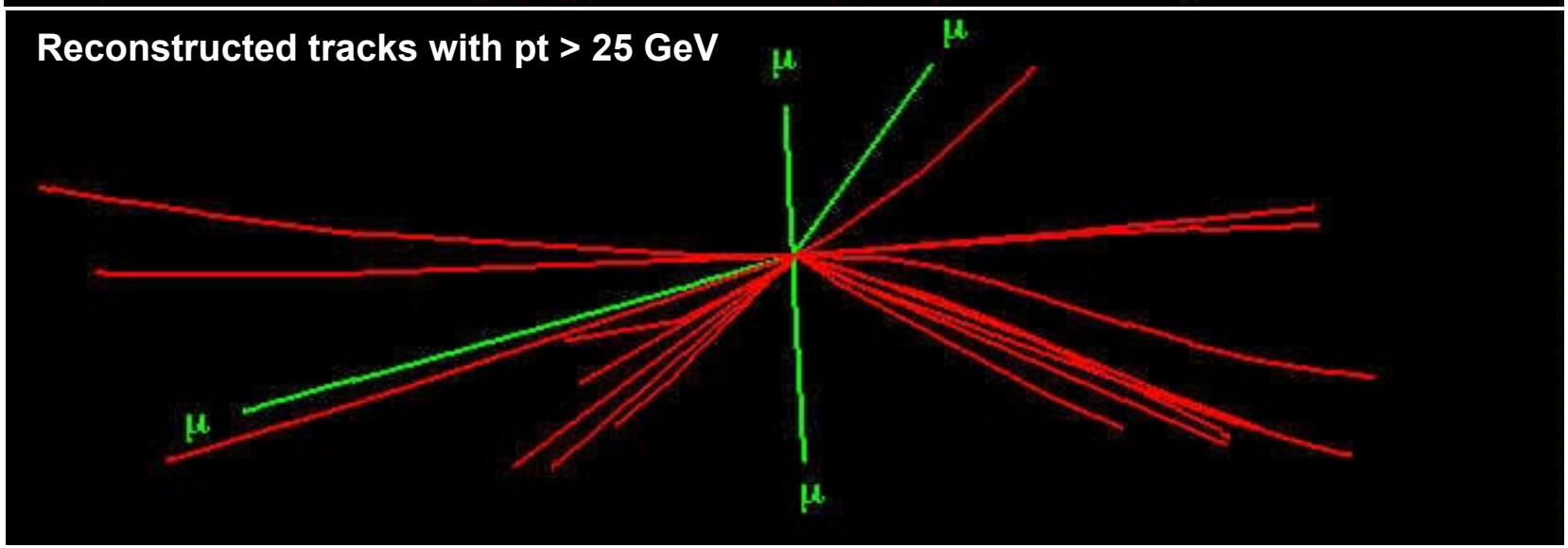


2007

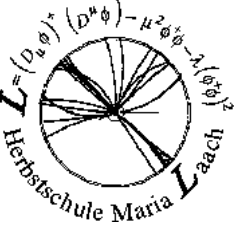


Example: Higgs Decay into 4 Muons

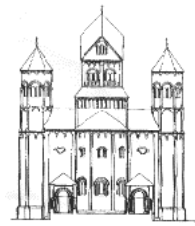




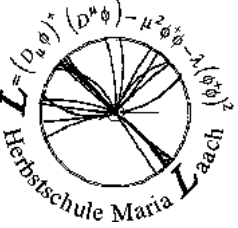
40M events/sec, selectivity: 1 in 10^{13}



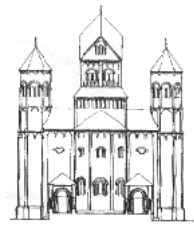
Software effort



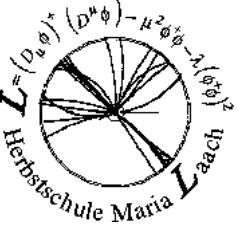
- The Core Software:
  - infrastructure which is required by detector-specific and analysis-specific software
  - includes items like the 'event-loop' and associated control, database infrastructure, calibration infrastructure, data-management infrastructure etc.
    - complexity of LHC software requires a well-engineered architecture and infrastructure
    - written by software experts in order that the Collaborations may contribute efficiently and that the result is manageable.
  - 'Core' does not include things like detector-specific reconstruction code or calibration code etc., nor of course the software of specific physics analyses (e.g. a neural-net analysis).



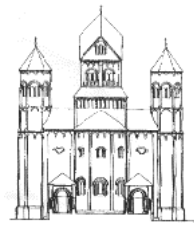
Software effort (2)



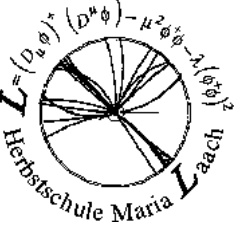
- The Core Software teams will be composed of software professionals and expert physicists coming from the LHC community.
- They will not necessarily be located at CERN, nor in any single place.
- The team will write the software infrastructure to enable to collect, to calibrate, to reconstruct, and to distribute the data.



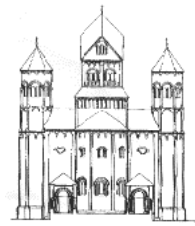
Core Software developers needed



Year	2000 have (missing)	2001	2002	2003	2004	2005
ALICE	12(5)	17.5	16.5	17	17.5	16.5
ATLAS[1]	23(8)	36	35	30	28	29
CMS	15(10)	27	31	33	33	33
LHCb	14(5)	25	24	23	22	21
Totals	**64(28)**	**105.5**	**106.5**	**103**	**100.5**	**99.5**



Software effort (3)



- The physics analysis programs, the detector-specific programs, calibration, etc…,
  - written by physicists.
- The four collaborations are confident that they will find the human resources to build the physics analysis software and to perform the physics analysis in connection with the Tier1 and Tier2 centres.

LHC: Event rate + Storage size

Parameter	Unit	ALICE p-p	ALICE Pb-Pb	ATLAS	CMS	LHCb	TOTAL		ATLAS (**)
Event recording rate	Hz	100	50	100	100	200			270
RAW Event size	MB	1	25	1	1	0.125			2
REC/ESD Event size	MB	0.1	2.5	0.5	0.5	0.1			0.5
AOD Event size	kB	10	250	10	10	20			10
TAG Event size	kB	1	10	0.1	1	1			0.1
Running time per year	M seconds	10	1	10	10	10			10
Events/year	Giga	1	0.05	1	1	2			2.7
Storage for real data	PB	1.2	1.5	2	1.7	0.45	6.9		8.1
RAW SIM Event size	MB	0.5	600	2	2	0.2			2
REC/ESD SIM Event size	MB	0.1	5	0.5	0.4	0.1			0.5
Events SIM/year	Giga	0.1	0.0001	0.12	0.5	1.2			0.12
Number of reconst. passes	Nb	2							
Storage for simul. data	PB	0.1	0.1	1.5	1.2	0.36	3.2		1.5
Storage for calibration	PB	0	0	0.4	0.01	0.01	0.4		0.4

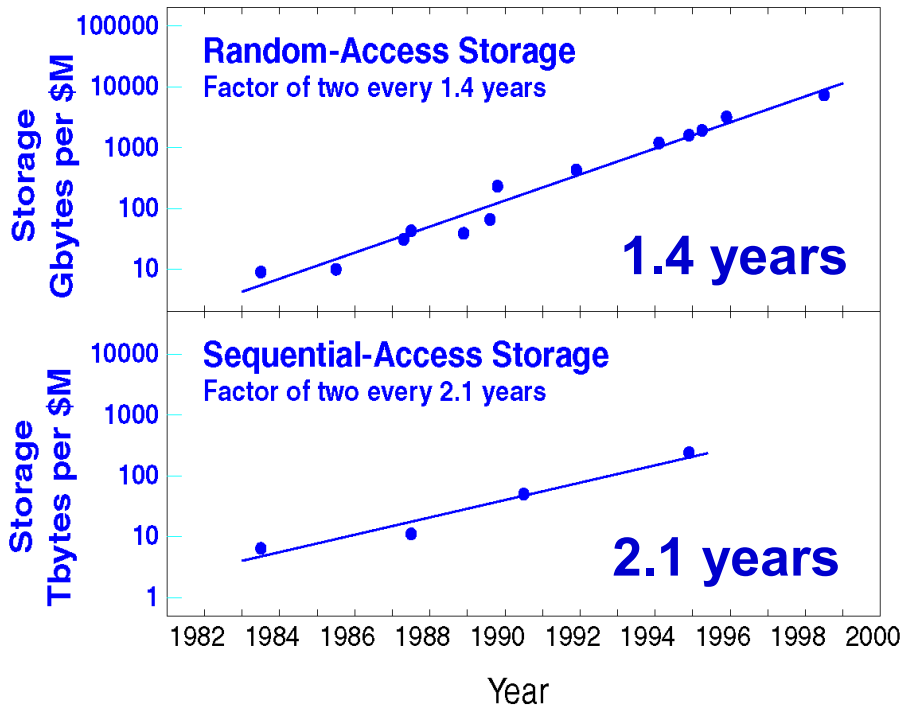
LHC: Tape/Disk Storage, Network

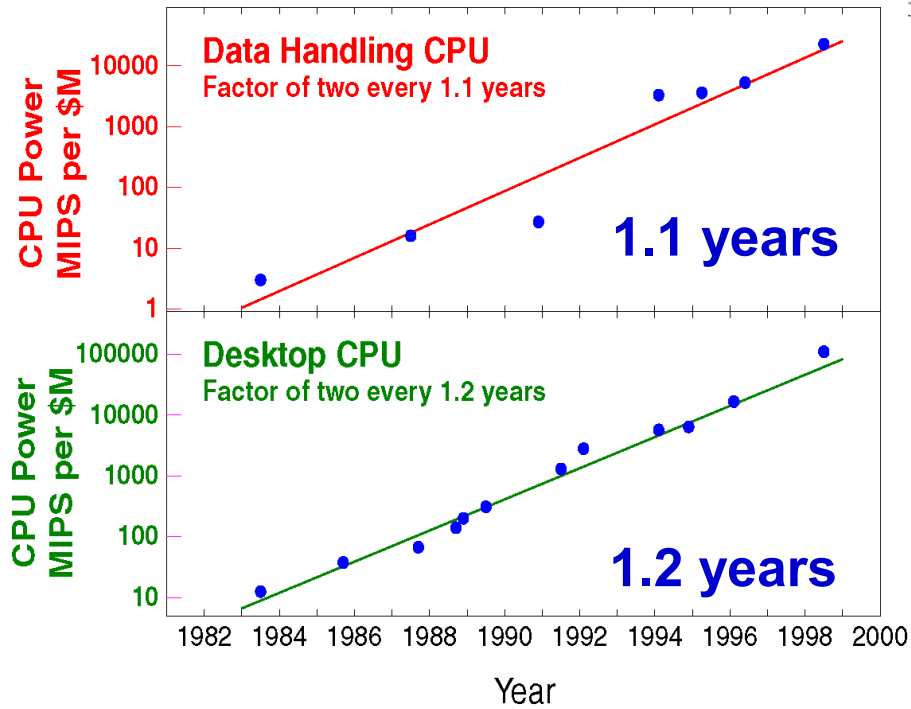
Parameter	Unit	ALICE p-p	ALICE Pb-Pb	ATLAS	CMS	LHCb	TOTAL		ATLAS (**)
# assumed Tier1 not at CERN		4		6	5	5			6
# assumed Tier2 not at CERN					25				
Tape storage at CERN T0+T1		3.23		2.86	4.17	1.22	11.5		9
Tape storage at each Tier1 (Avg.)	PB	0.37		1.26	1.02	0.32	3		1.8
Tape storage at each Tier2 (Avg.)	(10**15 B)				0.05				
∑ Tape storage / year		**4.7**		**10.4**	**10.5**	**2.8**	**28.5**		**19.8**
Disk storage at CERN T0+T1	PB	0.53		0.31	1.14	0.33	2.3		0.41
Disk storage at each Tier1 (Avg.)	PB	0.27		0.26	0.44	0.15	1.1		0.36
Disk storage at each Tier2 (Avg.)	PB				0.1				
∑ Disk storage	PB	**1.6**		**1.9**	**5.9**	**1.1**	**10.4**		**2.57**
WAN, Bandwidths									
Tier0 - Tier1 link, 1 expt.	Mbps	1500		1500	1500	310	4810		1500
Tier1 - Tier1 link	Mbps								
Tier1 - Tier2 link	Mbps	622		622	622				622

LHC: CPU requirements

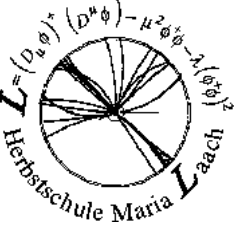
Parameter	Unit	ALICE p-p	ALICE Pb-Pb	ATLAS	CMS	LHCb	TOTAL		ATLAS (**)
Event recording rate	Hz	100	50	100	100	200			270
RAW Event size	MB	1	25	1	1	0.125			2
Time to reconstruct 1 event	k SI-95 sec	0.4	100	0.64	3	0.25			0.64
Time to simulate 1 event	k SI-95 sec	3	2250	3	5	1.5			3
CPU for 1 rec. pass/y (real data)	k SI-95	20	250	200	434	50			385
CPU for 1 SIM pass/y (sim+rec)	k SI-95	19	269	30	200	660			30
CPU reconstruction, calib.	k SI-95	65	525	251	1040	50	1931		435
CPU simulation	k SI-95	19	269	30	587	660	1564		30
CPU analysis	k SI-95	880		1479	1280	215	3854		1479
Total CPU at CERN T0+T1	k SI-95	824		506	820	225	2375		690
Total CPU each Tier1 (Avg.)	k SI-95	234		209	204	140	787		209
Total CPU each Tier2 (Avg.)	k SI-95				43				
∑ CPU	k SI-95	**1758**		**1760**	**2907**	**925**	**7349**		**1944**

Hardware Cost Estimates

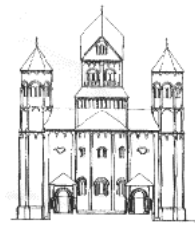




- Buy late, but not too late: phased implementation
  - R&D Phase 2001-2004
  - Implementation Phase 2004-2007
  - R&D to develop capabilities and computing model itself
  - Prototyping at increasing scales of capability & complexity



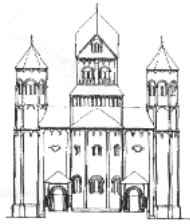
Cost of CERN-based LHC computing



Overall hardware and tape costs of the initial CERN-based Computing Facility constructed in the years 2005-2007 (kCHF)

	ALICE	ATLAS	CMS	LHCb
CPU	11'069	10'667	12'667	3'479
Disk Pool	2'188	1'907	5'314	1'535
Automated Tape	3'200	9'407	1'617	958
Shelf Tape	0	0	1'816	214
Tape I/O	1'616	1'711	1'711	855
Total cost	**18'073**	**23'692**	**23'135**	**7'040**
%CPU Cost	61.2%	45.0%	54.8%	49.4%
%Disk Pool Cost	12.1%	8.0%	23.0%	21.8%
%Automated Tape Cost	17.7%	39.7%	7.0%	13.6%
%Shelf Tape Cost	0.0%	0.0%	7.0%	3.0%
%Tape I/O Cost	8.9%	7.2%	7.4%	12.2%

Resources at CERN vs. elsewhere



- Substantial computing and analysis resources will be located and deployed at regional centers
  - This calls for a new analysis model:
    "Many of you will not do your analysis at CERN"

	ALICE	ATLAS	CMS	LHCb	Total
#Tier1	4	6	5	5	
Avg. Tier1 [kCHF]	7'095	6'835	13'638	4'030	
All Tier1's [kCHF]	28'381	41'012	68'189	20'152	157'733
CERN [kCHF]	18'073	23'692	23'135	7'040	71'940
CERN/Total	39%	37%	25%	26%	31%

- For comparison:
  - BaBar investment at SLAC (1997-200): 35MCHF
  - Run2 computing investment at FNAL (1998-2002): 30MCHF

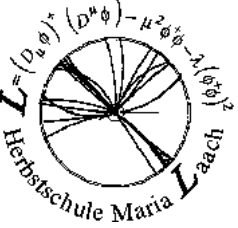
Computing** for High Energy Physics

33. Herbstschule für Hochenergiephysik
4.-14.9.2001

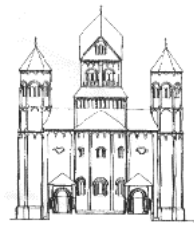
Matthias Kasemann
Fermilab

Part 3

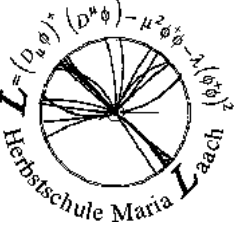
**Computing == Computing and Analysis



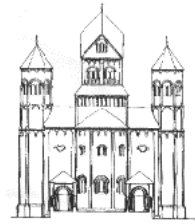
LHC computing: challenges



- perform data challenges of increasing size and complexity
- Current cost estimates based on forecast evolution of price and performance of computer hardware
- hardware costs of initial set-up of LHC distributed computer centres (Tier-0 to -2):
  - 240 MCHF
  - CERN-based Tier-0+1 centre: about 1/3 of total.
- investment for initial system to be spent in 2005, 2006 and 2007, in ~ equal portions
  - (assuming LHC start-up in 2006 and reach of design luminosity in 2007)
- Materials & Operation of LHC computing system:
  - rolling replacement within constant budget
  - requires ~ 1/3 of initial investment per year (~ 80 MCHF world-wide) - includes steady evolution of capacity
- set-up of a common prototype as joint project (experiments, CERN-IT, major regional centres),
  - reaching ~50% of overall computing structure of 1 LHC experiment by ~2003/4

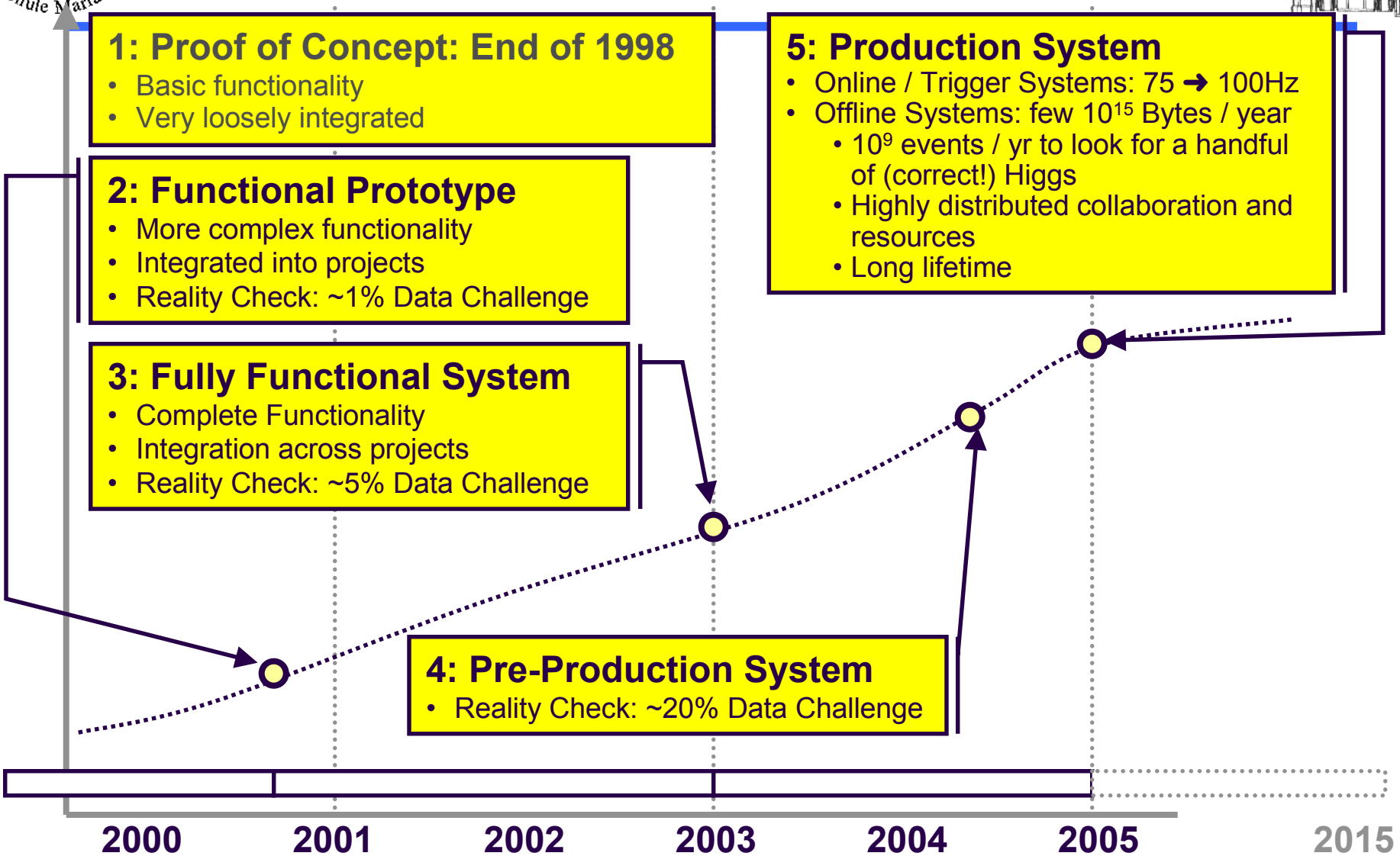


CMS Core Software and Computing Project Scope



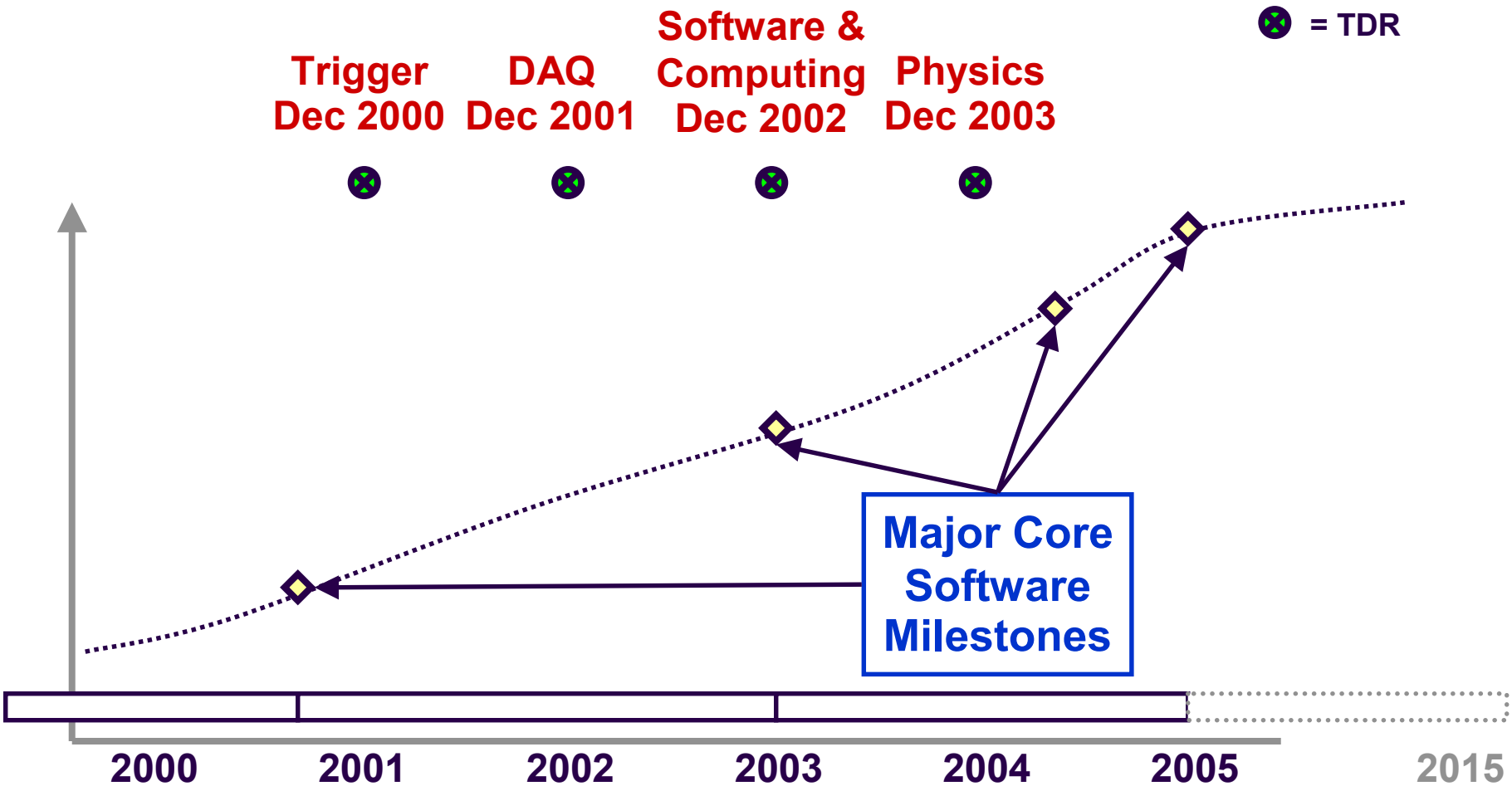
- Framework, Architecture, Tools and Facilities for
  - Design, evaluation and calibration of the detector
  - Storage, access, distribution and processing of data
  - Event simulation, reconstruction and analysis
  - Distributed collaboration, software development, data processing, and physics analysis
- An integral part of CMS and crucial to its success
  - Now, similar to a subdetector system in terms of scale and complexity
  - Treated organizationally in CMS as any subdetector system
  - Will be a main activity of CMS during LHC operation

Software Development Phases



Significant Requirements from CMS TDR's

or: "it's not just an evolution to 2005 software"



CMS Milestones: Software

- CMS software development strategy:
  - First, completed transition to C++, now functionality with good performance, then optimized performance

CMS MILESTONES	1998	1999	2000	2001	2002	2003	2004	2005
CORE SOFTWARE								
End of Fortran development	Jun-98							
GEANT4 simulation of CMS	1 Jun-98		2 Dec-99		3 Jun-01		4 Dec-03	
Reconstruction/analysis framework	1 Jun-98		2 Dec-99		3 Dec-01		4 Dec-03	
Detector reconstruction		1 Dec-98	2 Dec-99			3 Jun-02		4 Jun-04
Physics object reconstruction		1 Mar-99		2 Jun-00		3 Dec-02		4 Dec-04
User analysis environment		1 Dec-98		2 Jun-00		3 Dec-02		4 Dec-04

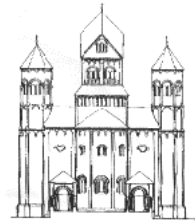
1 Proof of concept
2 Functional prototype
3 Fully functional
4 Production system

In 2005, need fully functional, tested, high quality, performing software

Phases: test ideas, make prototypes, develop modules, and integrate

In 2000, we have functional prototypes
Next, create the basis for final system

CMS: Current Analysis Chain



Prod.

- CMSIM (Geant3) Simulation
- ORCA Hit formatting into ODBMS
- ORCA Pileup and Digitization of some subset of detectors

Users

- Selection of events for further processing
- Create new collection
  - shallow (links), or deep (data) to existing objects
  - add new objects (ie. Tk Digits)
  - replace existing objects if required
- All combinations of
  - Transient - (Persistent)
  - Persistent - Transient
  - Persistent - Transient - Persistent

User collections of events of interest

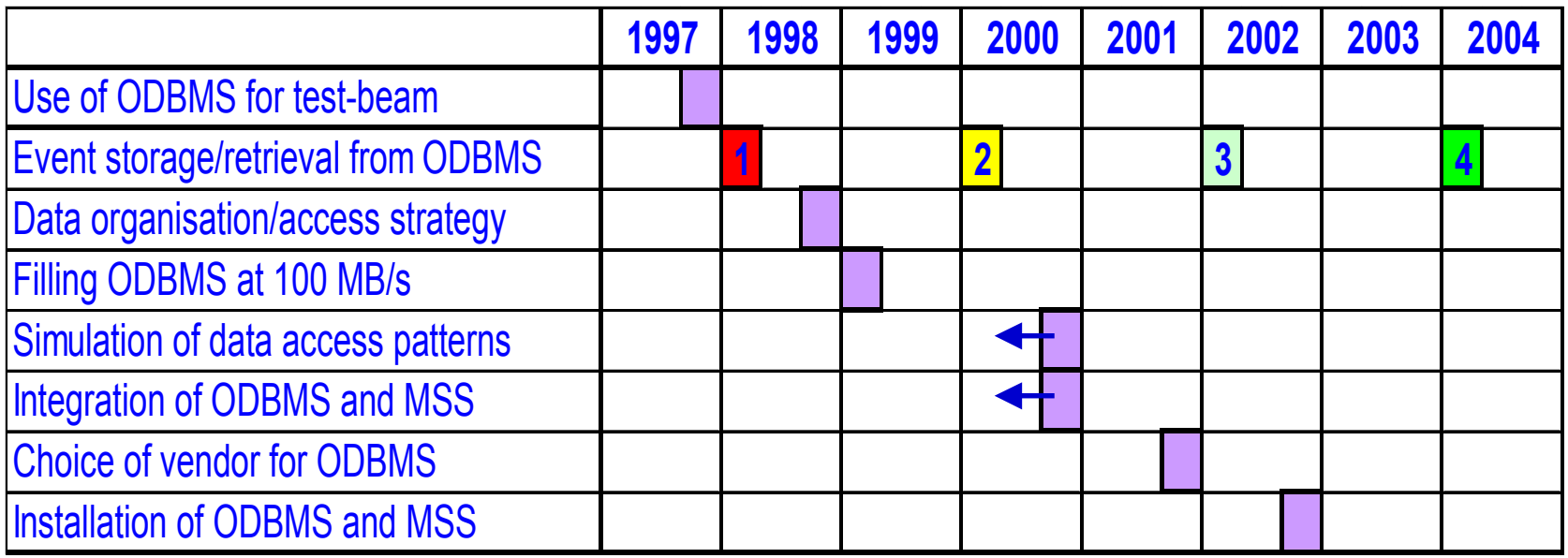
- Collections can span datasets, navigation back from event data to run data can be used to get correct setups

Full ODBMS functionality being used

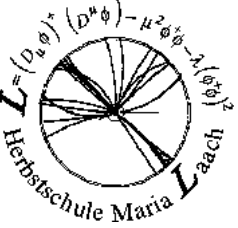
Organization of the Meta-data and production issues, are typically much more technically complex than the simple issues of persistent objects!

CMS Milestones: Data

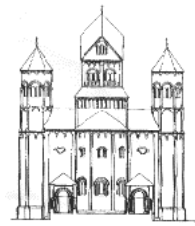
- ~ 10 Petabytes for raw data, simulated data, reconstructed data, physics objects, calibration data, user data, ...



Need to add GRID milestones for distributed system integration

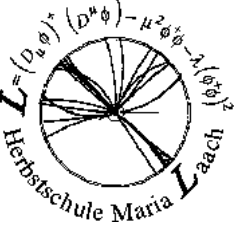


CMS: Data Handling

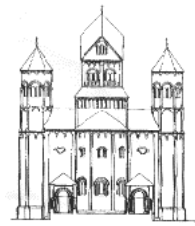


- We already have more data (10TB) than disk space
  - use MSS systems (HPSS at CERN, ENSTORE at FNAL)
  - automatic staging by hooks in Objectivity code
- Tools are in place to replicate federations
  - Shallow (catalog and schema files only)
  - Meta-deep (plus meta-data)
  - Deep (plus Database files)
  - LAN and WAN
- "Request Redirection Protocols" being prototyped and field tested
  - allows to "hide" actual data location from users
    - disk goes down, single change on a central server redirects user requests to a new server
    - A powerful place to hook into Objectivity to give us the required control

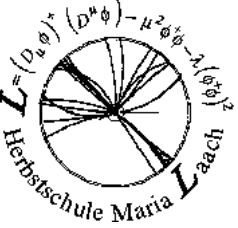
Leverage products like Objectivity to reduce the amount of SW we have to write



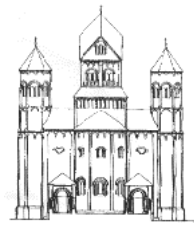
CMS: 5% and 20% Data Challenges



- In 2002/2004 we have 5% and 20% data challenges specified
  - In fact these are just part of a steady process
- The % we refer to is the % of complexity, rather than the genuine 5% of the data, or 5% of the CPU…
- Use Farms sized (in boxes) at 5/20%
  - processing speed whatever it is
  - run for a period of a month or so to see the whole system in operation:
    - first pass reconstruction
    - roll data out to users continuously (no scheduled downtimes)
    - selected "streams" (collections) in operation
    - user offline selection -> user collections
    - replication between sites
    - timely "results"

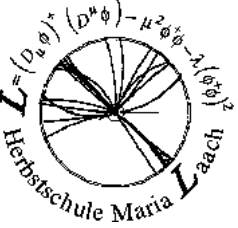


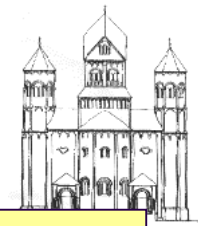
CMS Computing Solution: A Data Grid



- Deploy computing resources as hierarchical grid
  - Tier 0 ⇒ Central laboratory computing resources (CERN)
  - Tier 1 ⇒ National center (Fermilab / BNL, other countries)
  - Tier 2 ⇒ Regional computing center (university)
  - Tier 3 ⇒ University group computing resources
  - Tier 4 ⇒ Individual workstation/CPU
- We call this arrangement a “Data Grid” to reflect the overwhelming role that data plays in deployment
- LHC data volume / Current experiments: factor 2-4
  - CDF: ~ 450 TB/year
  - Compass: ~ 300 TB/year of RAW data
  - STAR: ~ 200 TB/year of RAW data

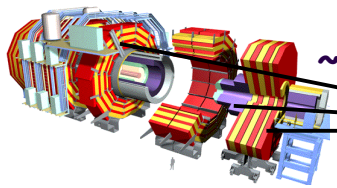
Example: CMS Data Grid





CERN/Outside Resource Ratio ~1:2
Tier0/(Σ Tier1)/(Σ Tier2) ~1:1:1

Experiment



~PBytes/sec

Online System

~100 MBytes/sec

Bunch crossing per 25 nsecs.
100 triggers per second
Event is ~1 MByte in size

Tier 0 +1

CERN Computer Center > 20 TIPS

2.5 Gbits/sec

France Center

UK Center

Italy Center

USA Center

Tier 1

2.5 Gbits/sec

Tier 2

Tier2 Center

~622 Mbits/sec

Tier 3

Institute ~0.25TIPS

Physics data cache

100 - 1000 Mbits/sec

Tier 4

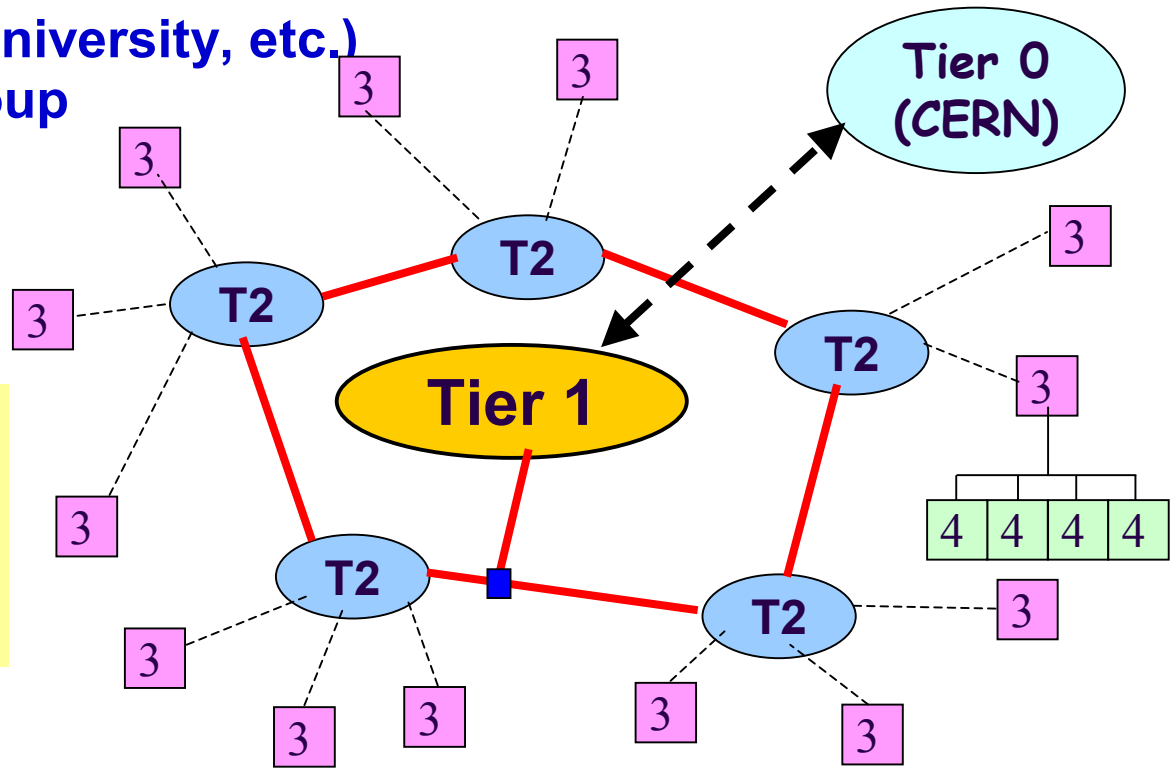
Workstations, other portals

Physicists work on analysis "channels".

Each institute has ~10 physicists working on one or more channels

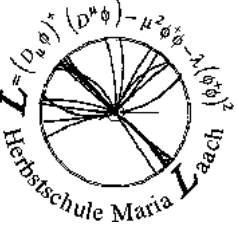
Global LHC Data Grid Hierarchy

Tier0 CERN
Tier1 National Lab
Tier2 Regional Center (University, etc.)
Tier3 University workgroup
Tier4 Workstation

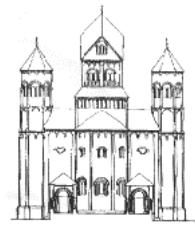


Key ideas:

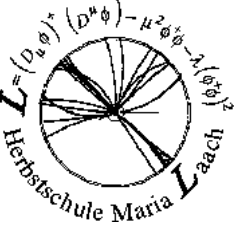
- **Hierarchical structure**
- **Tier0-1-2 centers**
- **Operate as unified Grid**



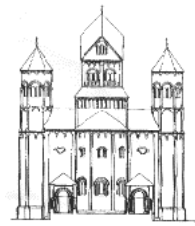
Tier1 and Tier2 Centers



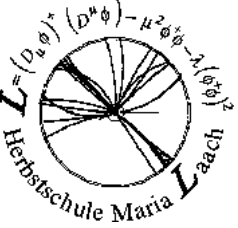
- Tier1 centers
  - National laboratory scale: large CPU, disk, tape resources
  - High speed networks
  - Many personnel with broad expertise
  - Central resource for large region
- Tier2 centers
  - New concept in LHC distributed computing hierarchy
  - Size ≈ [national lab * university]$^{1/2}$
  - Based at large University or small laboratory
  - Emphasis on small staff, simple configuration & operation
- Tier2 role
  - Simulations, analysis, data caching
  - Serve small country, or region within large country



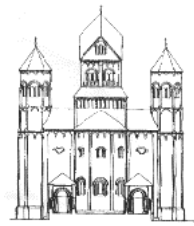
Why Regional Centers?



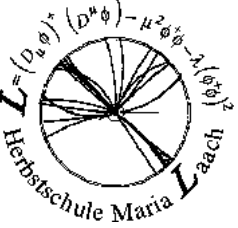
- Bring computing facilities closer to home
  - final analysis on a compact cluster in the physics department
- Exploit established computing expertise & infrastructure
- Reduce dependence on links to CERN
  - full ESD available nearby - through a fat, fast, reliable network link
- Tap funding sources not otherwise available to HEP
- Devolve control over resource allocation
  - national interests?
  - regional interests?
  - at the expense of physics interests?



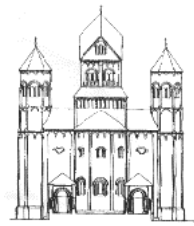
Regional Centers Services and Facilities



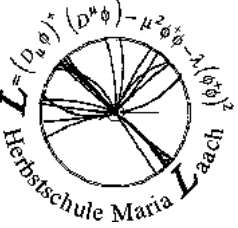
- Regional Centers will
  - Provide all technical services and data services required to do the analysis
  - Maintain all (or a large fraction of) the processed analysis data. Possibly may only have large subsets based on physics channels. Maintain a fixed fraction of fully reconstructed and raw data
  - Cache or mirror the calibration constants
  - Maintain excellent network connectivity to CERN and excellent connectivity to users in the region. Data transfer over the network is preferred for all transactions but transfer of very large datasets on removable data volumes is not ruled out.
  - Share/develop common maintenance, validation, and production software with CERN and the collaboration



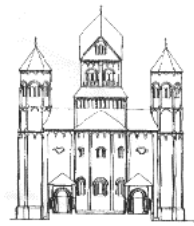
Regional Centers Services and Facilities



- Provide services to physicists in the region, contribute a fair share to post-reconstruction processing and data analysis, collaborate with other RCs and CERN on common projects, and provide services to members of other regions on a best effort basis to further the science of the experiment
- Provide support services, training, documentation, trouble shooting to RC and remote users in the region

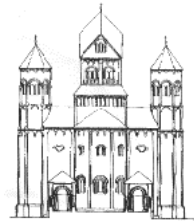


Motivations for Regional Centers



- To maximize the intellectual contribution of physicists all over the world without requiring their physical presence at CERN
- Acknowledgement of possible limitations of network bandwidth
- A way of utilizing the expertise and resources residing in computing centers all over the world
- Allows people to make choices on how they analyze data based on availability or proximity of various resources such as CPU, data, or network bandwidth.

The MONARC RC Topology



Tier 1

University physics department
- Final analysis
- Dedicated to local users
- Limited data capacity – cached only via the network
- Zero administration costs (fully automated)

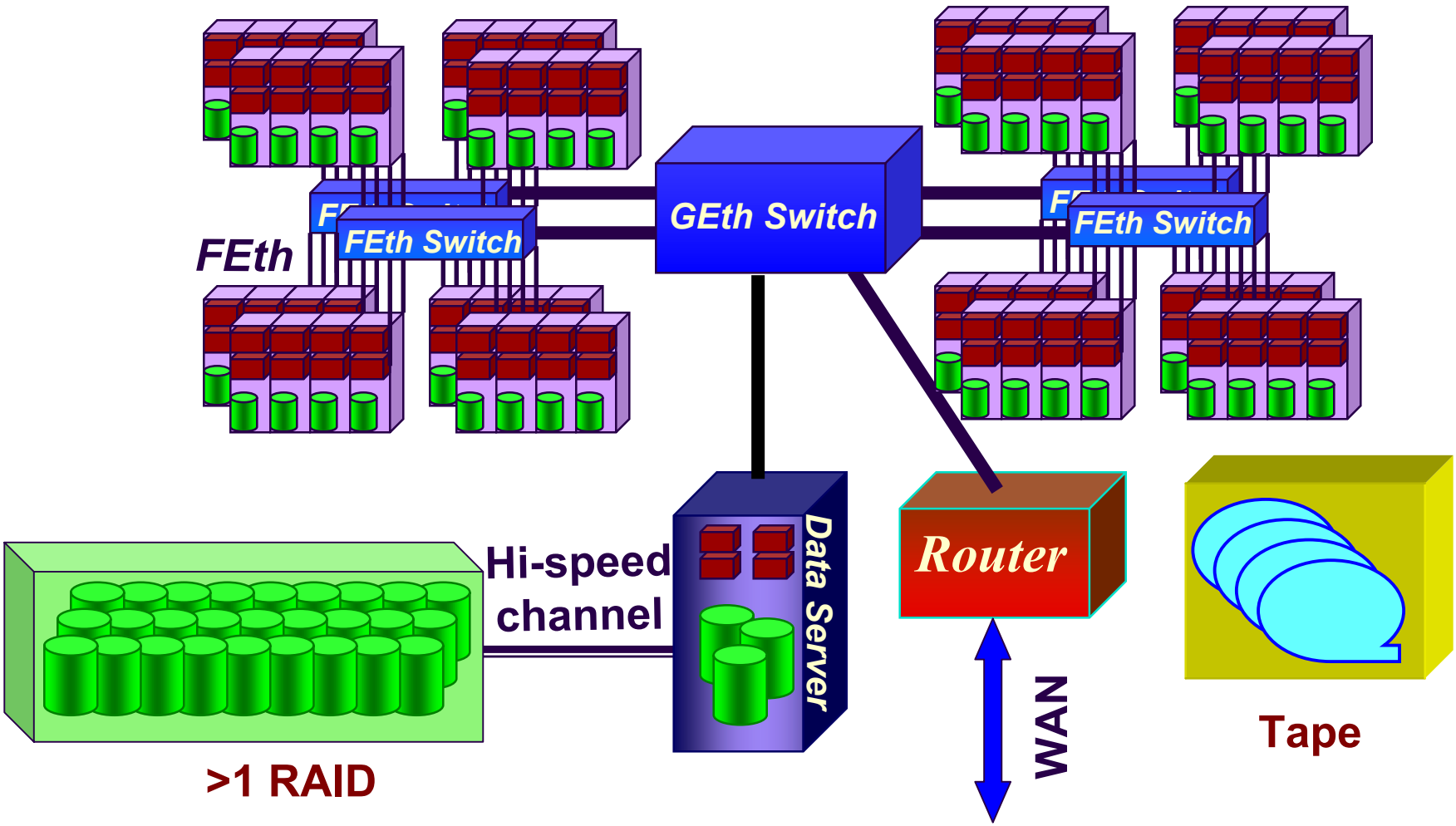
Tier2

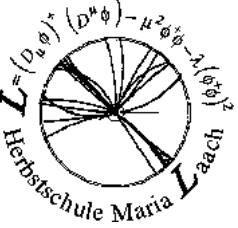
Tier 1 – established data centre
or new facility hosted by a lab
- Major subset of data – all/most of the ESD, selected raw da
- Mass
- ESD
- Fat
- High
- User

Depart

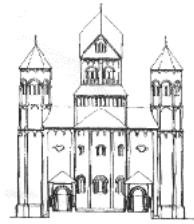
Tier 2 – smaller labs, smaller countries,
probably hosted by existing data cen
- Mainly AOD analysis
- Data cached from Tier 1, Tier 0 centres
- No mass storage management
- Minimal staffing costs

LHC Tier2 Centre (2001)





The full costs?



- Space
- Power, cooling
- Software

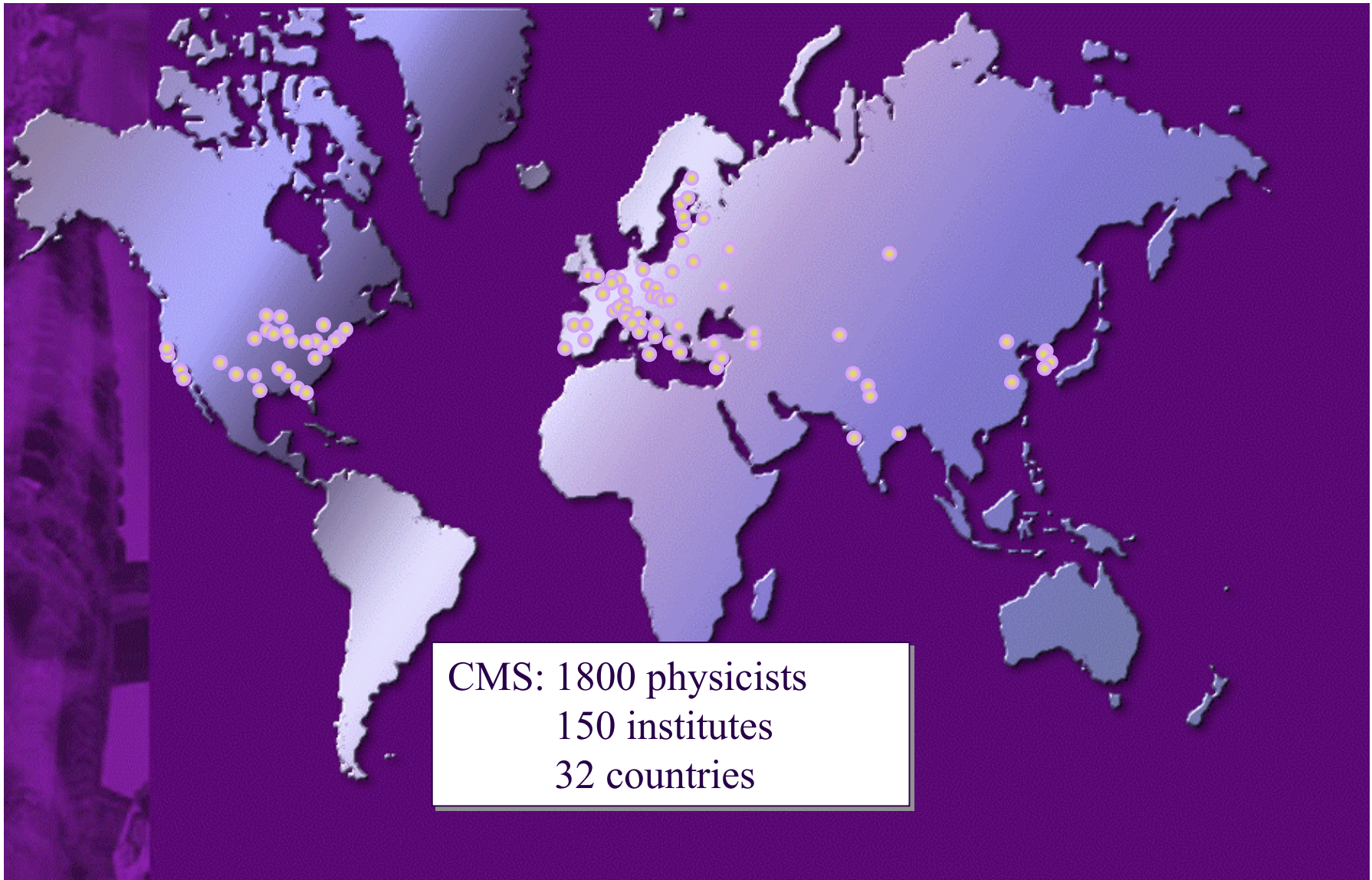
- LAN
- Replacement/Expansion 30% per year

- Mass storage

- People

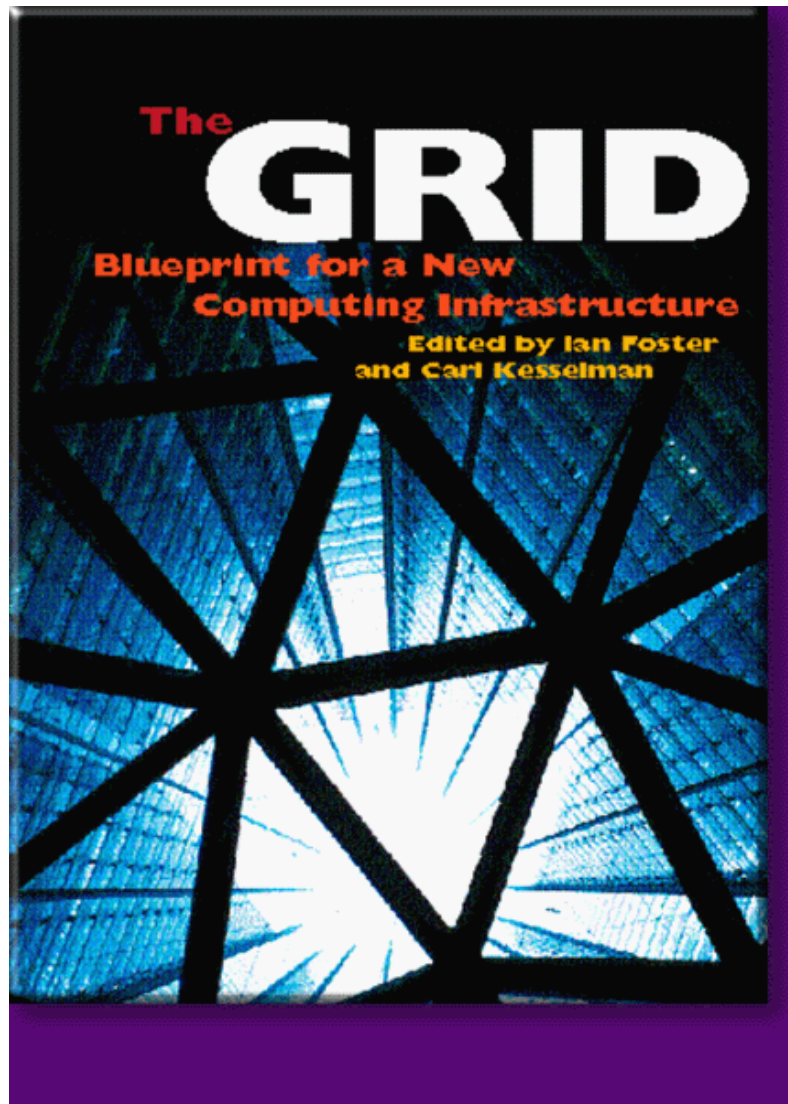
World Wide Collaboration

⇒ distributed computing & storage capacity

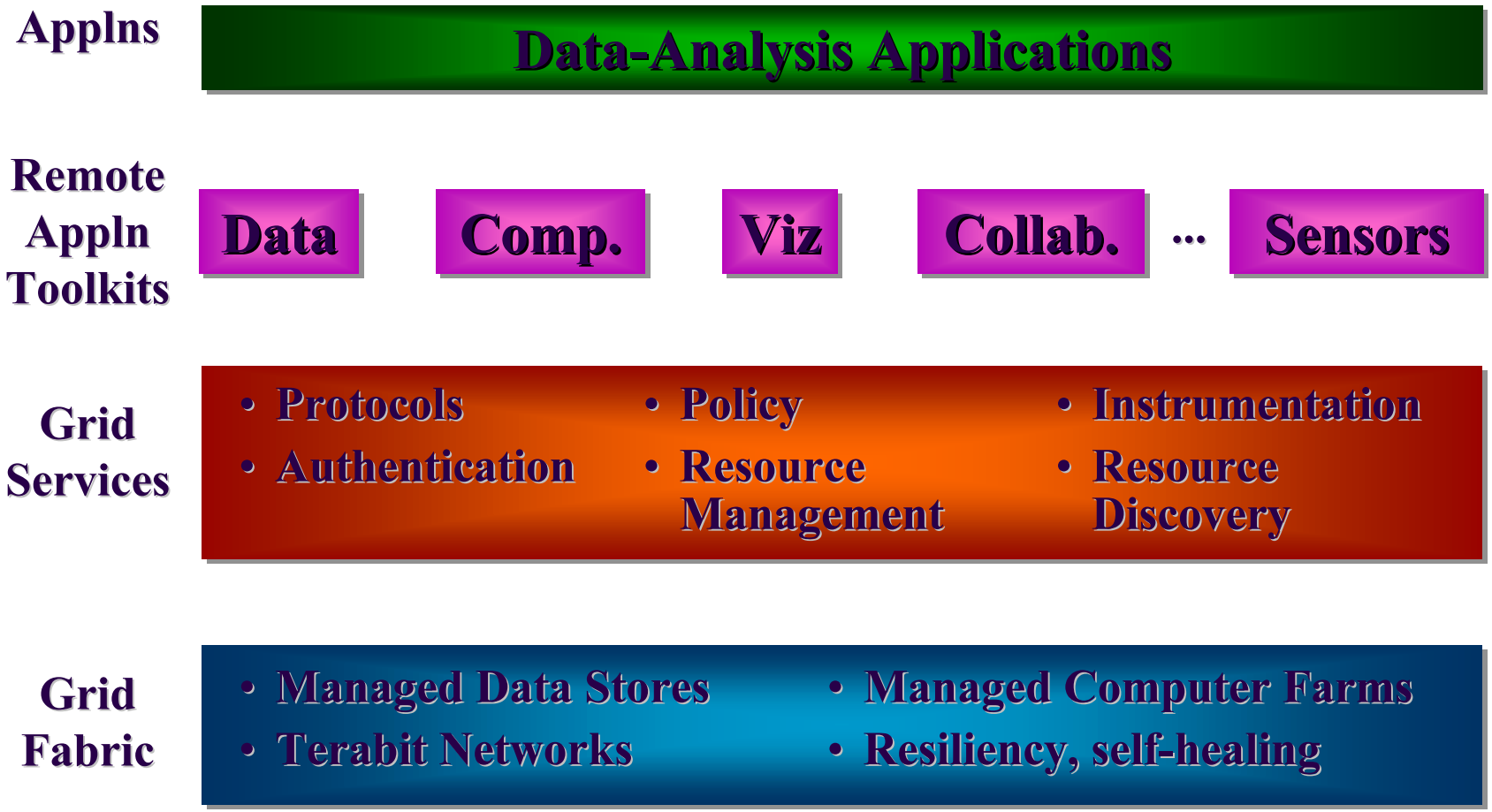


Information Grids: the solution to the LHC Data Challenge ?

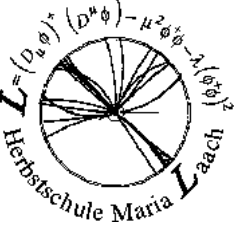
- Next step after Web/Internet
- Information Sockets dynamically deliver data and computational resources
- Analogy to the Electric Grid
- Major difference:
  All electrons are similar…
  All bits of information are not.
- Hot research topic



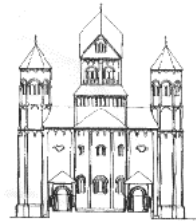
Grid Services Architecture *



* Adapted from Ian Foster: computing, data and access (collaborative) grids,...

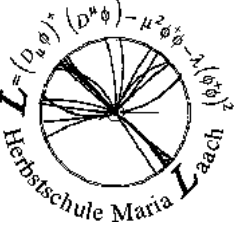


The Fundamental Concept

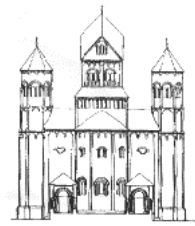


Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

Enable communities (“virtual organizations”) to share geographically distributed resources as they pursue common goals—in the absence of central control, omniscience, trust relationships



One View of Requirements



Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

- Identity & authentication
- Authorization & policy
- Resource discovery
- Resource characterization
- Resource allocation
- (Co-)reservation, workflow
- Distributed algorithms
- Remote data access
- High-speed data transfer
- Performance guarantees
- Monitoring
- Adaptation
- Intrusion detection
- Resource management
- Accounting & payment
- Fault management
- System evolution
- Etc.
- Etc.
- …

Another View: "Three Obstacles to Making Grid Computing Routine"

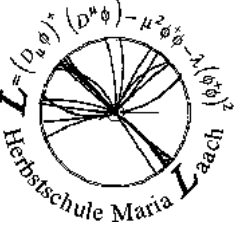
Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

1) New approaches to problem solving
   - Data Grids, distributed computing, peer-to-peer, collaboration grids, …
2) Structuring and writing programs
   - Abstractions, tools

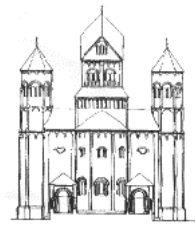
Programming Problem

3) Enabling resource sharing across distinct institutions
   - Resource discovery, access, reservation, allocation; authentication, authorization, policy; communication; fault detection and notification; …

Systems Problem

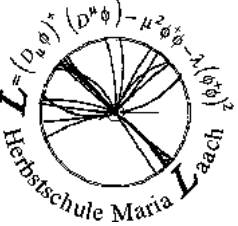


Aspects of the Systems Problem

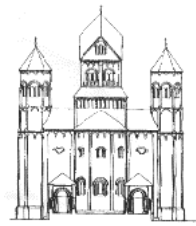


Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

1) Need for interoperability when different groups want to share resources
   - Diverse components, policies, mechanisms
   - E.g., standard notions of identity, means of communication, resource descriptions
2) Need for shared infrastructure services to avoid repeated development, installation
   - E.g., one port/service/protocol for remote access to computing, not one per tool/appln
   - E.g., Certificate Authorities: expensive to run
- A common need for protocols & services



Protocol-Oriented View of Grid Architecture



Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

- Development of Grid protocols & services
  - Protocol-mediated access to remote resources
  - New services: e.g., resource brokering
  - "On the Grid" = speak Intergrid protocols
  - Mostly (extensions to) existing protocols
- Development of Grid APIs & SDKs
  - Facilitate application development by supplying higher-level abstractions
- The (hugely successful) model is the Internet

Layered Grid Architecture (By Analogy to Internet Architecture)

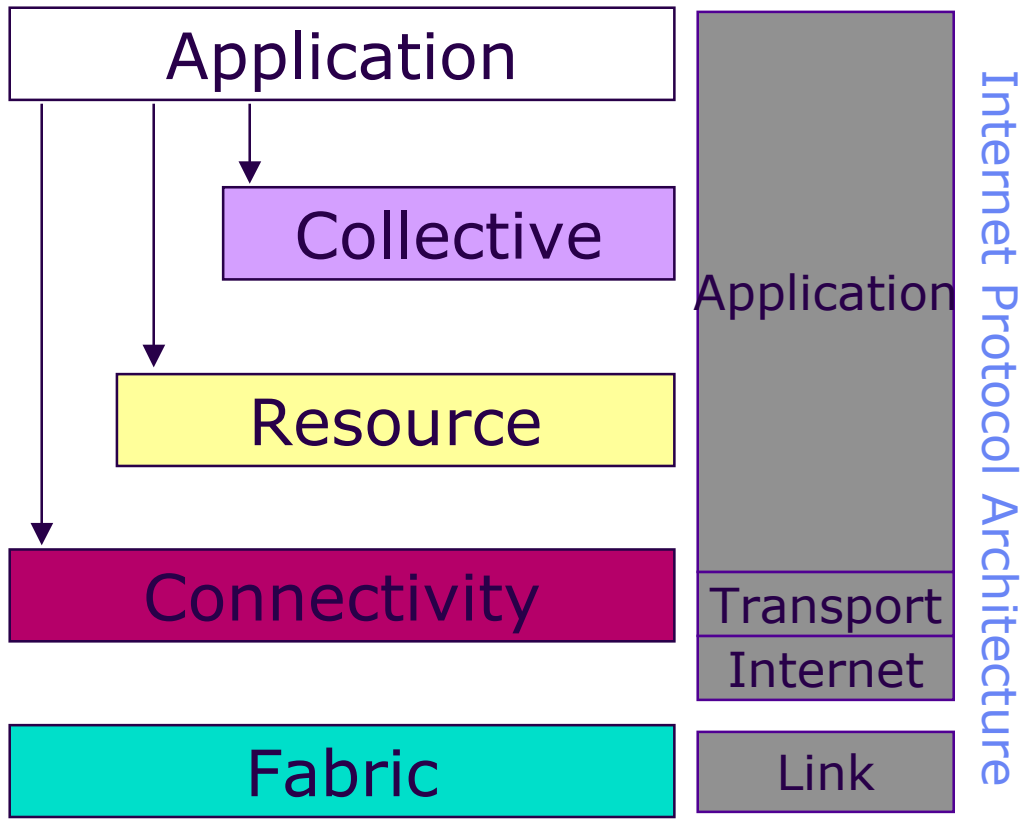
Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

"Coordinating multiple resources": ubiquitous infrastructure services, app-specific distributed services

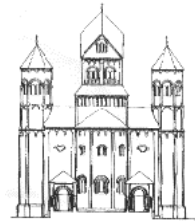
"Sharing single resources": negotiating access, controlling use

"Talking to things": communication (Internet protocols) & security

"Controlling things locally": Access to, & control of, resources



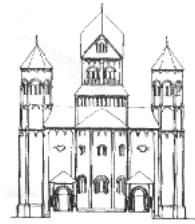
Globus Toolkit



Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

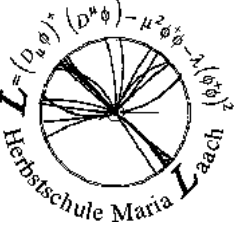
- Globus Toolkit is the source of many of the protocols described in “Grid architecture”
- Adopted by almost all major Grid projects worldwide as a source of infrastructure
- Open source, open architecture framework encourages community development
- Active R&D program continues to move technology forward
- Developers at ANL, USC/ISI, NCSA, LBNL, and other institutions

Globus Toolkit Components Include …

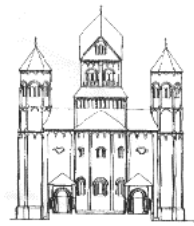


Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

- Core protocols and services
  - Grid Security Infrastructure
  - Grid Resource Access & Management
  - MDS information & monitoring
  - GridFTP data access & transfer
- Other services
  - Community Authorization Service
  - DUROC co-allocation service
- Other Data Grid technologies
  - Replica catalog, replica management service

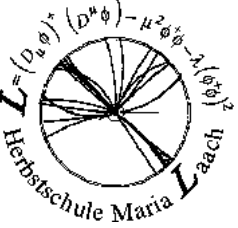


The Programming Problem

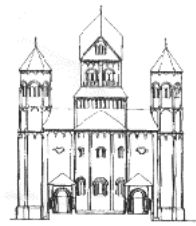


Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

- How the @#$@ do I develop robust, secure, long-lived applications for dynamic, heterogeneous, Grids?
- I need, presumably:
  - Abstractions and models to add to speed/robustness/etc. of development
  - Tools to ease application development and diagnose common problems
  - Code/tool sharing to allow reuse of code components developed by others

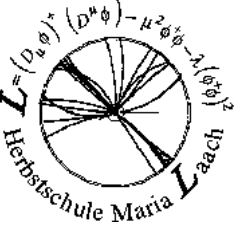


Grid Programming Technologies

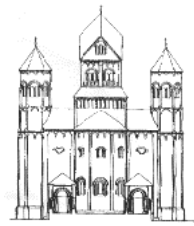


Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

- "Grid applications" are incredibly diverse (data, collaboration, computing, sensors, …)
  - Seems unlikely there is one solution
- Most applications have been written "from scratch," with or without Grid services
- Application-specific libraries have been shown to provide significant benefits
- No new language, programming model, etc., has yet emerged that transforms things
  - But certainly still quite possible

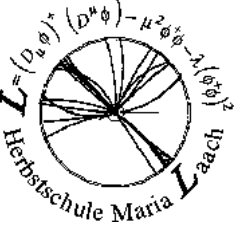


Examples of Grid Programming Technologies

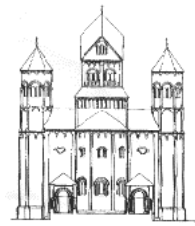


Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

- MPICH-G2: Grid-enabled message passing
- CoG Kits, GridPort: Portal construction, based on N-tier architectures
- GDMP, Data Grid Tools, SRB: replica management, collection management
- Condor-G: simple workflow management
- Cactus: Grid-aware numerical solver framework
  - Note tremendous variety, application focus



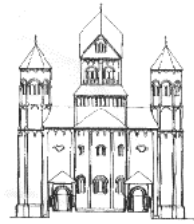
Globus Applications and Deployments

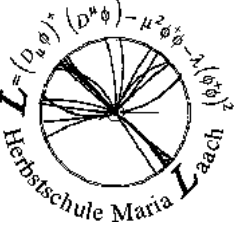


Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

- Application projects include
  - GriPhyN, PPDG, NEES, EU DataGrid, ESG, Fusion Collaboratory, etc., etc.
- Infrastructure deployments include
  - DISCOM, NASA IPG, NSF TeraGrid, DOE Science Grid, EU DataGrid, etc., etc.
  - UK Grid Center, U.S. GRIDS Center
- Technology projects include
  - Data Grids, Access Grid, Portals, CORBA, MPICH-G2, Condor-G, GrADS, etc., etc.

Data Grid Reference Architecture





Application

Discipline- Specific Data Grid Applications

Collective

- Consistency Management Services
- Usage Accounting Services
- Request Management Services
- Request Planning Services
- Replica Selection Services
- Replica Management Services
- System Monitoring Services
- Resource Brokering Services
- Information Services
- Coallocation Services
- Distributed Catalog Services
- Community Authorization Service
- Online Certificate Repository

Resource

- Storage Mgmt Protocol
- Compute Mgmt Protocol
- Network Mgmt Protocol
- Catalog Mgmt Protocol
- Code Mgmt Protocol
- Service Reg. Protocol
- Enquiry Protocol

Connectivity

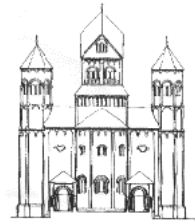
Communication, service discovery (DNS), authentication, delegation

Fabric

- Storage Systems
- Compute Systems
- Networks
- Catalogs
- Code Repositories

Are Grids a solution?

Les Robertson



Computational Grids

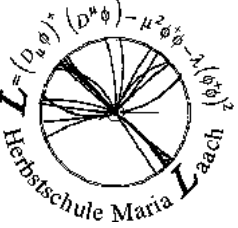
- Change of orientation of Meta-computing activity
  - From inter-connected super-computers … .. towards a more general concept of a computational power Grid (The Grid – Ian Foster, Carl Kesselman**)
- Has found resonance with the press, funding agencies

But what is a Grid?

"Dependable, consistent, pervasive access to resources**"

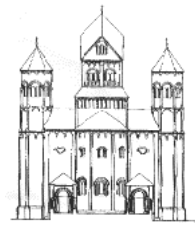
So, in some way Grid technology makes it easy to use diverse, geographically distributed, locally managed and controlled computing facilities – as if they formed a coherent local cluster

** Ian Foster and Carl Kesselman, editors, "The Grid: Blueprint for a New Computing Infrastructure," Morgan Kaufmann, 1999

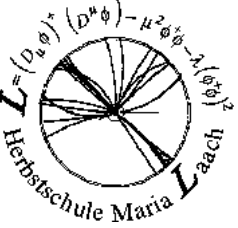


What does the Grid do for you?

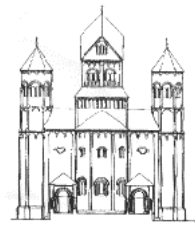
Les Robertson



- You submit your work
- And the Grid
  - Finds convenient places for it to be run
  - Organises efficient access to your data
    - Caching, migration, replication
  - Deals with authentication to the different sites that you will be using
  - Interfaces to local site resource allocation mechanisms, policies
  - Runs your jobs
  - Monitors progress
  - Recovers from problems
  - Tells you when your work is complete
- If there is scope for parallelism, it can also decompose your work into convenient execution units based on the available resources, data distribution

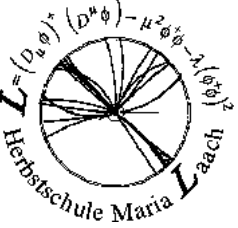


Example Application Projects

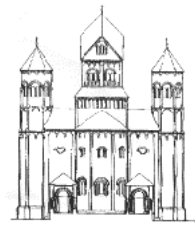


Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

- AstroGrid: astronomy, etc. (UK)
- Earth Systems Grid: environment (US DOE)
- EU DataGrid: physics, environment, etc. (EU)
- EuroGrid: various (EU)
- Fusion Collaboratory (US DOE)
- GridLab: astrophysics, etc. (EU)
- Grid Physics Network (US NSF)
- MetaNEOS: numerical optimization (US NSF)
- NEESgrid: civil engineering (US NSF)
- Particle Physics Data Grid (US DOE)

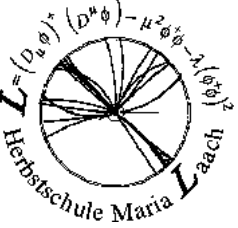


"A Rich Technology Base has been Constructed"

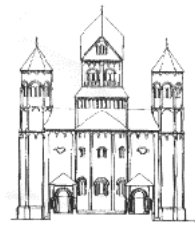


Carl Kesselman
Center for Grid Technologies
USC/Information Sciences Institute

- 6+ years of R&D have produced a substantial code base based on open architecture principles: esp. the Globus Toolkit, including
  - Grid Security Infrastructure
  - Resource directory and discovery services
  - Secure remote resource access
  - Data Grid protocols, services, and tools
- Essentially all major projects have adopted this as a common suite of protocols & services
- Enabling wide range of higher-level services

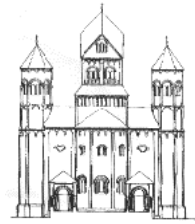


HEP Related Data Grid Projects



- Funded projects
  - GriPhyN USA NSF, $11.9M + $1.6M
  - PPDG I USA DOE, $2M
  - PPDG II USA DOE, $9.5M
  - EU DataGrid EU $9.3M
- Proposed projects
  - iVDGL USA NSF, $15M + $1.8M + UK
  - DTF USA NSF, $45M + $4M/yr
  - DataTag EU EC, $2M?
  - GridPP UK PPARC, > $15M
- Other national projects
  - UK e-Science (> $100M for 2001-2004)
  - Italy, France, (Japan?)

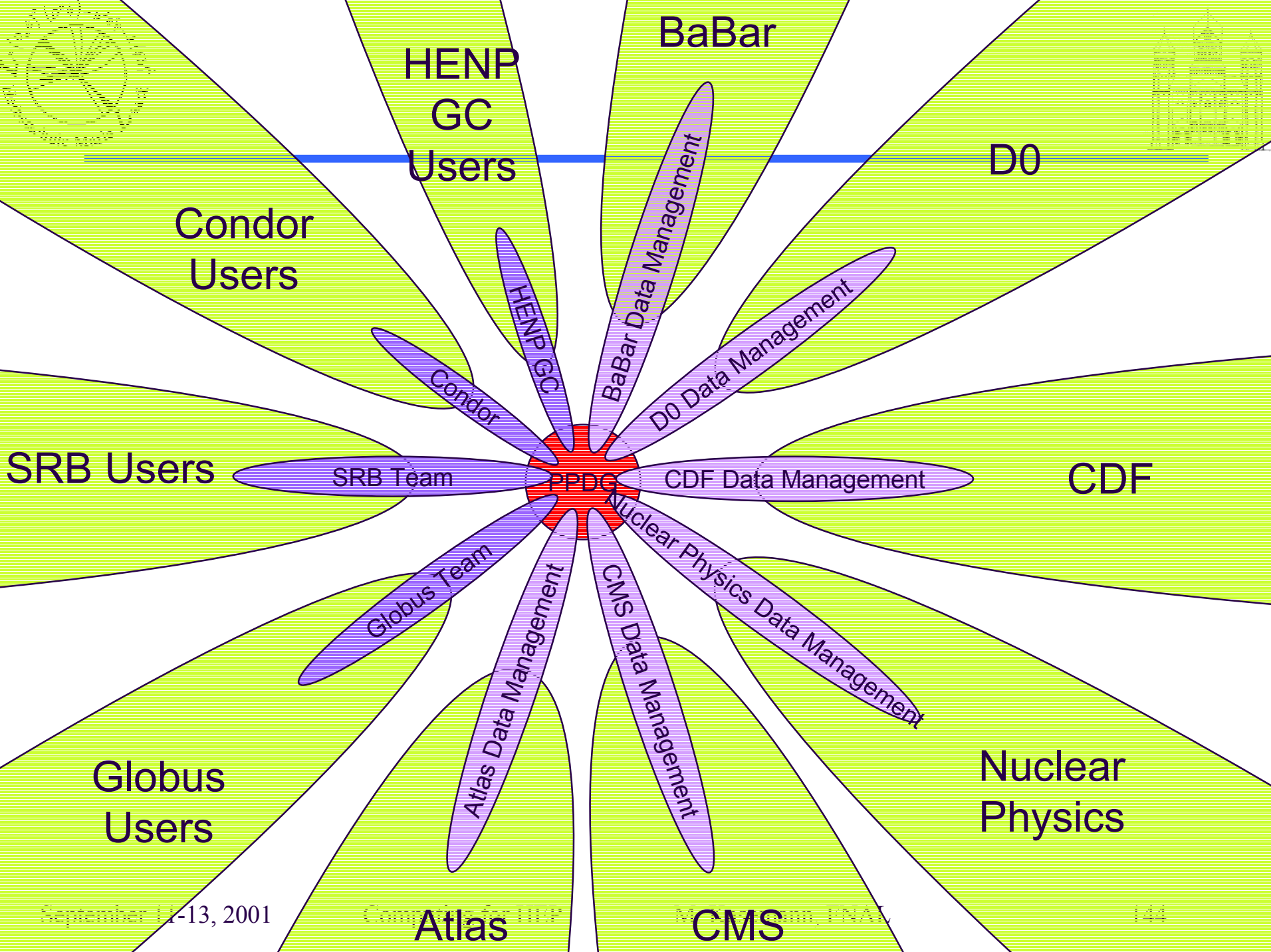
Coordination Among Grid Projects

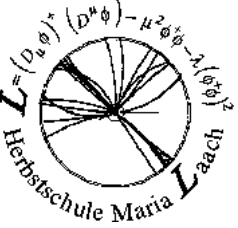


- Particle Physics Data Grid (US, DOE)
  - Data Grid applications for HENP
  - Funded 1999, 2000 ($2M)
  - Funded 2001-2004 ($9.4M)
  - http://www.ppdg.net/
- GriPhyN (US, NSF)
  - Petascale Virtual-Data Grids
  - Funded 9/2000 – 9/2005 ($11.9M+$1.6M)
  - http://www.griphyn.org/
- European Data Grid (EU)
  - Data Grid technologies, EU deployment
  - Funded 1/2001 – 1/2004 ($9.3M)

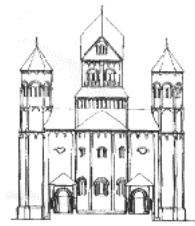
- HEP in common
- Focus: infrastructure development & deployment
- International scope
- Now developing joint coordination framework

GridPP, DTF, iVDGL ⇒ very soon?


HENP GC Users
BaBar
D0
Condor Users
SRB Users
CDF
Globus Users
Nuclear Physics
Atlas
CMS
HENP GC
BaBar Data Management
D0 Data Management
Condor
SRB Team
PPDG
CDF Data Management
Globus Team
Atlas Data Management
CMS Data Management
Nuclear Physics Data Management

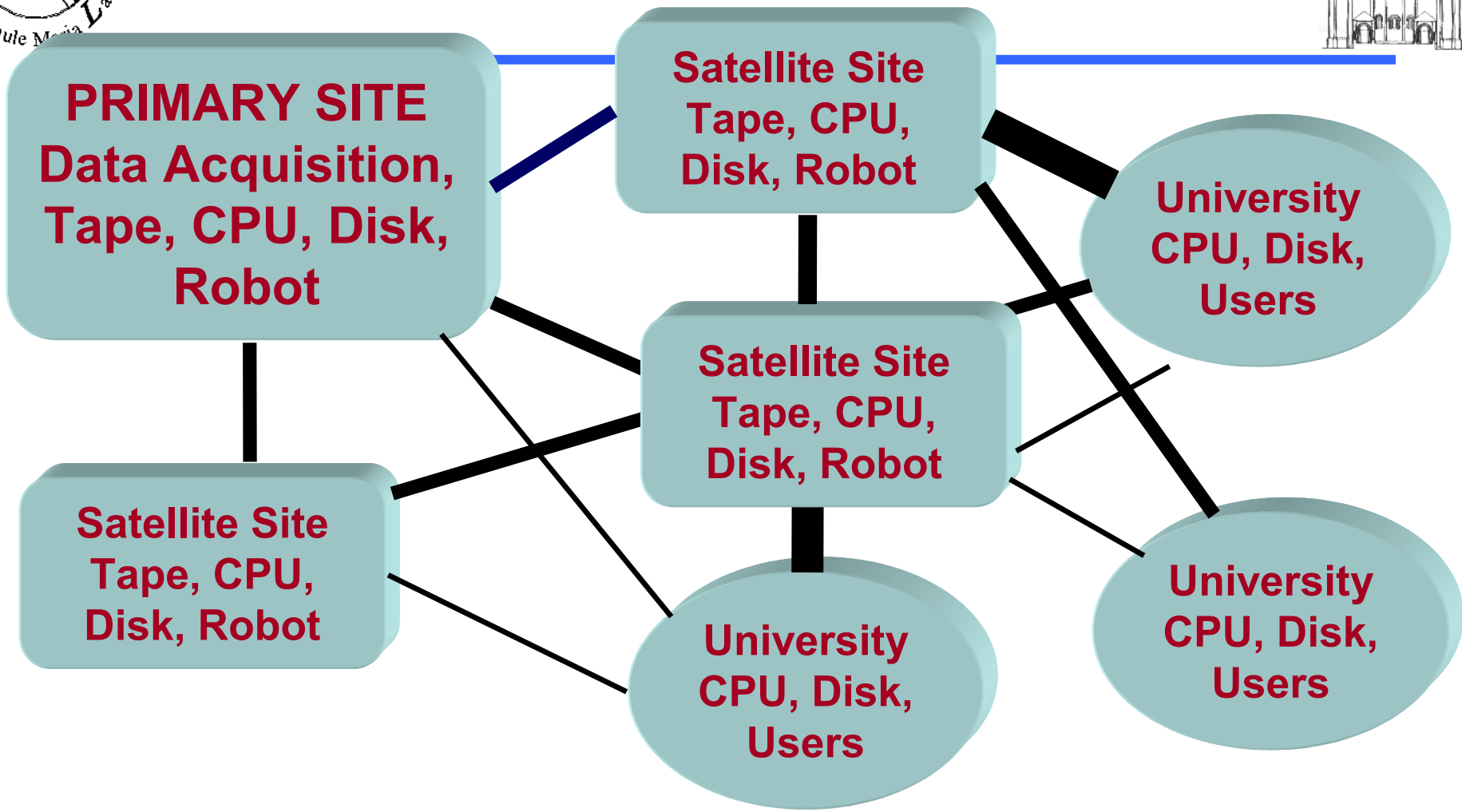


PPDG and GriPhyN Projects



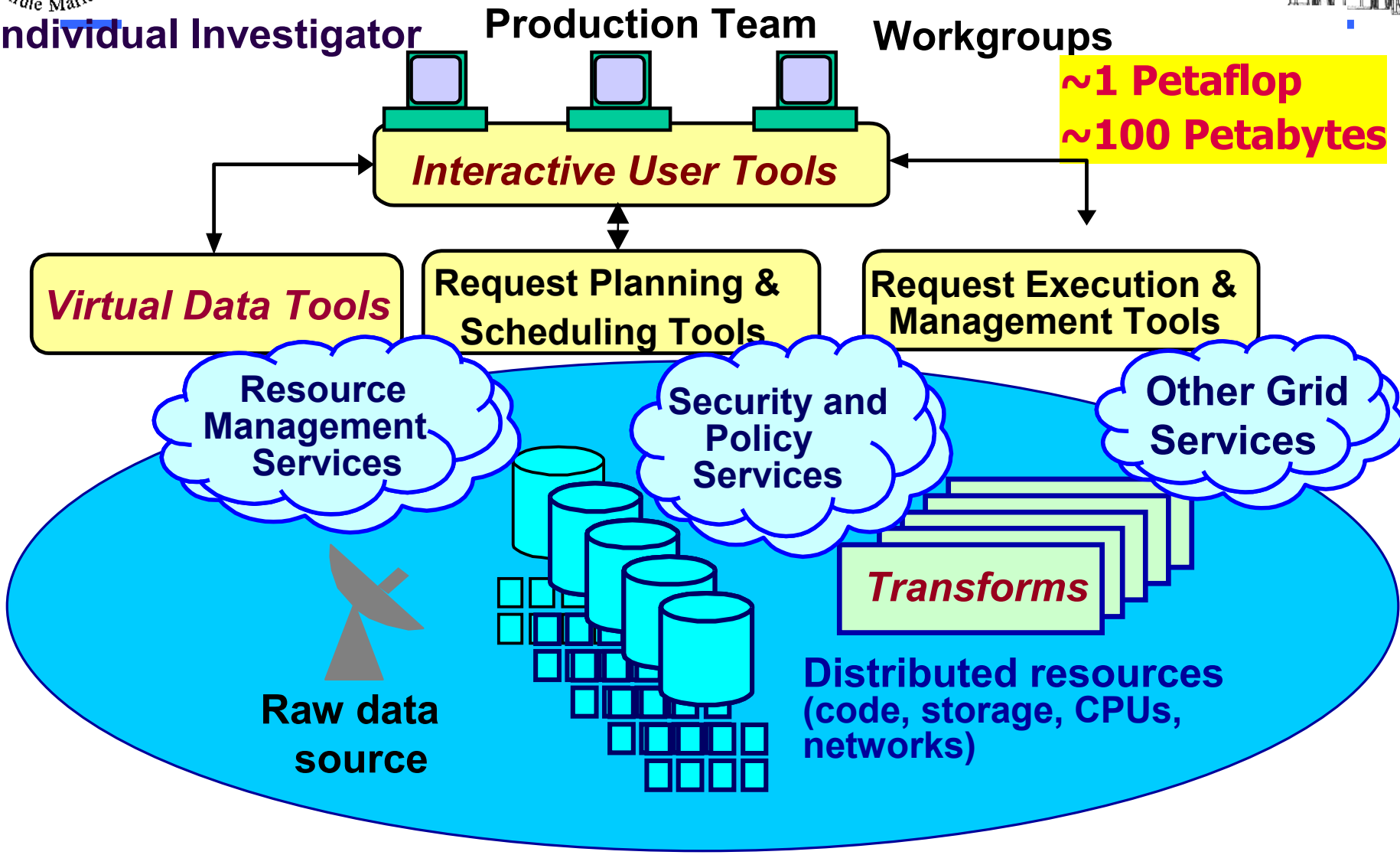
- PPDG focus on today's (evolving) problems in HENP
  - Current HEP: BaBar, CDF, D0
  - Current NP: RHIC, JLAB
  - Future HEP: ATLAS , CMS
- GriPhyN focus on tomorrow's solutions
  - ATLAS, CMS, LIGO, SDSS
  - Virtual data, "Petascale" problems (Petaflops, Petabytes)
  - Toolkit, export to other disciplines, outreach/education
- Both emphasize
  - Application sciences drivers
  - CS/application partnership (reflected in funding)
  - Performance
- Explicitly complementary

PPDG Multi-site Cached File Access System



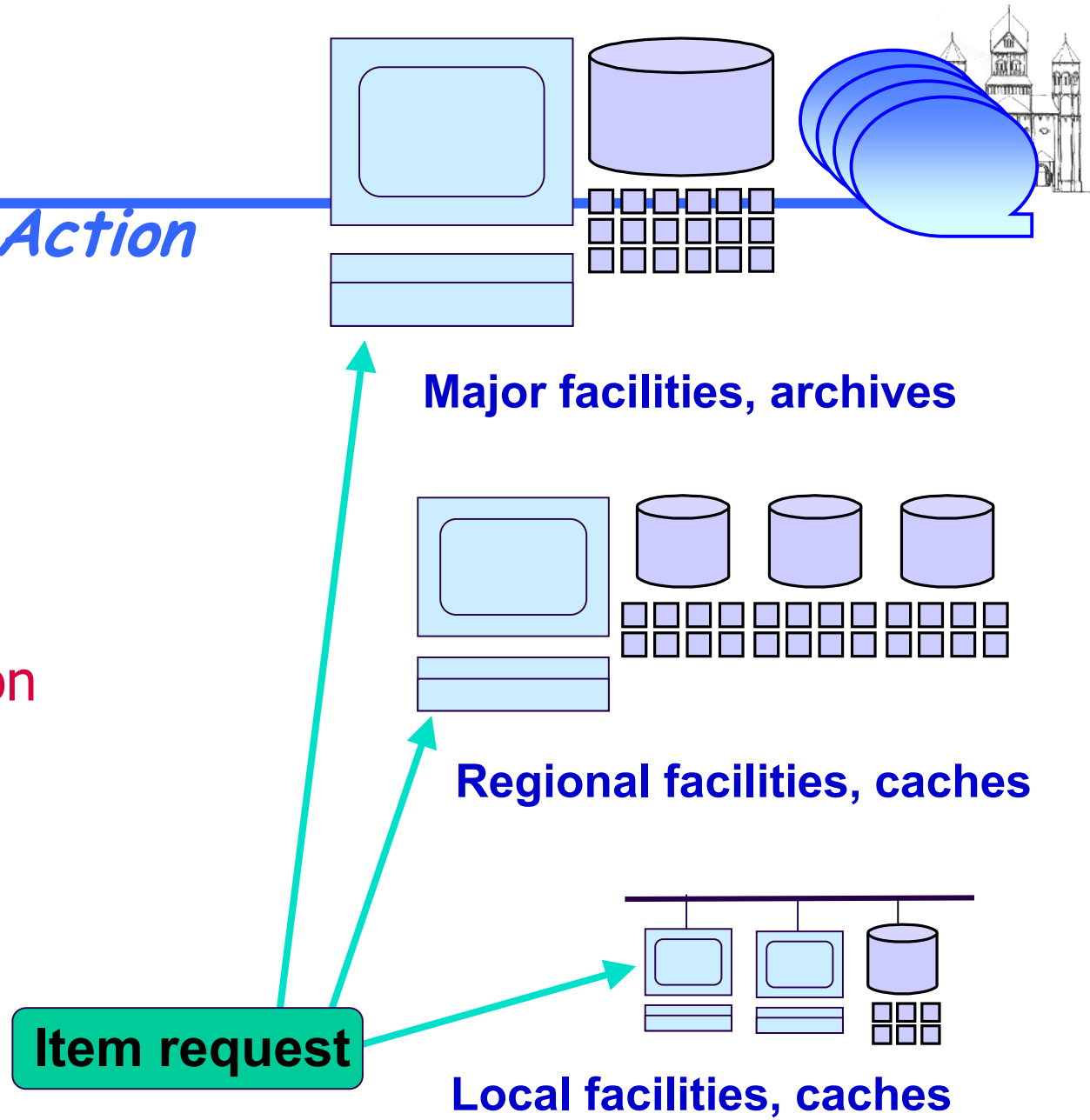
Resource Discovery, Matchmaking, Co-Scheduling/Queueing, Tracking/Monitoring, Problem Trapping + Resolution

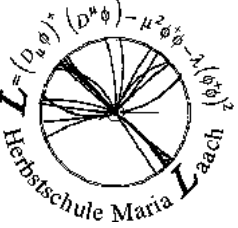
GriPhyN: PetaScale Virtual-Data Grids



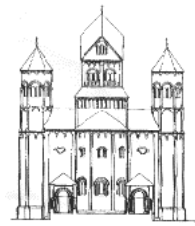
Virtual Data in Action

- Data request may
  - Compute locally
  - Compute remotely
  - Access local data
  - Access remote data
- Scheduling based on
  - Local policies
  - Global policies
  - Cost



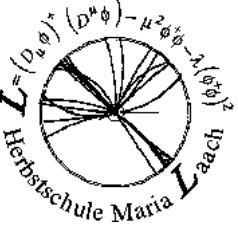


GriPhyN Goals for Virtual Data

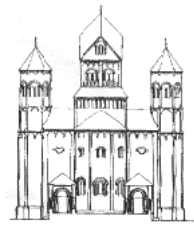


Explore concept of virtual data and its applicability to data-intensive science

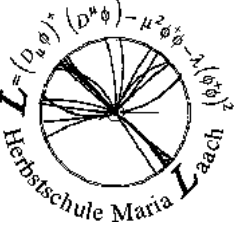
- Transparency with respect to location
  - Caching, catalogs, in a large-scale, high-performance Data Grid
- Transparency with respect to materialization
  - Exact specification of algorithm components
    - Traceability of any data product
  - Cost of storage vs CPU vs networks
- Automated management of computation
  - Issues of scale, complexity, transparency
  - Complications: calibrations, data versions, software versions, …



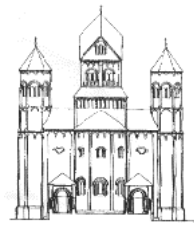
GriPhyN Overview (www.griphyn.org)



- 5-year, $12M NSF ITR proposal to realize the concept of virtual data, via:
  1) CS research on
     - Virtual data technologies (info models, management of virtual data software, etc.)
     - Request planning and scheduling (including policy representation and enforcement)
     - Task execution (including agent computing, fault management, etc.)
  2) Development of Virtual Data Toolkit (VDT)
  3) Applications: ATLAS, CMS, LIGO, SDSS
- PIs=Avery (Florida), Foster (Chicago)

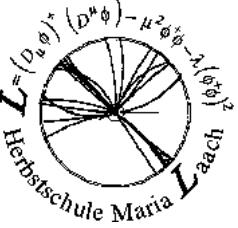


EU DataGrid Project

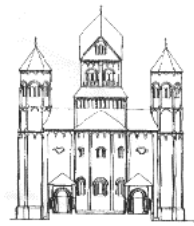


LHC/HEP focus

Work Package	Work Package title	Lead contractor
WP1	Grid Workload Management	INFN
WP2	Grid Data Management	CERN
WP3	Grid Monitoring Services	PPARC
WP4	Fabric Management	CERN
WP5	Mass Storage Management	PPARC
WP6	Integration Testbed	CNRS
WP7	Network Services	CNRS
WP8	High Energy Physics Applications	CERN
WP9	Earth Observation Science Applications	ESA
WP10	Biology Science Applications	INFN
WP11	Dissemination and Exploitation	INFN
WP12	Project Management	CERN



GRID Computing

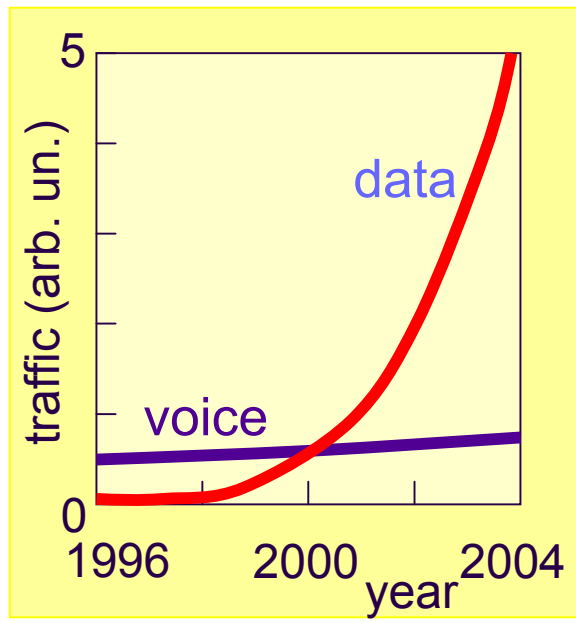


- GRID computing is a very hot topic at the moment.
- HENP is involved in many GRID R&D projects, with the next steps aimed at providing real tools and software to experiments.
- The problem is a large one and it is not yet clear that the concepts will turned into effective computing.
  - CMS@HOME (á la seti@home)
- Most Grid Projects initiated by Computer Science
  - addresses very HEP relevant questions
  - projects very much aligned with HEP projects
  - HEP serves as excellent test-bed

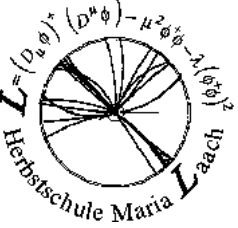
- We (HEP) need working implementations for our experiments!!
  - Better be involved, and we are
  - *"Try to lead, then you can set direction"*

Things taken for granted (II): internet

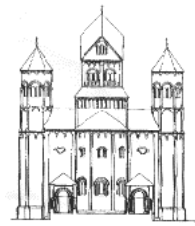
- 100 million new users expected online by 2001
  - Internet traffic is doubled every 100 days
  - 5000 domain names added every day
  - 1 Billion web pages (Inktomi & NEC Res. Inst.)
- Commerce in 2001: >$200M
- 1999: last year of the voice
- Prices(basic units) dropping
- Conclusion:
  - It'll go on; can count on it.



Pietro M. DI VITA / Telecom ITALIA
Telecom99



Data Storage/Access (I)



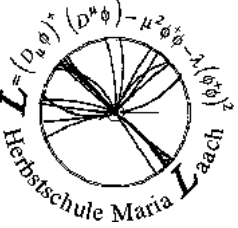
M. Shapiro

Persistent Event Data: To OODBMS or not to OODBMS

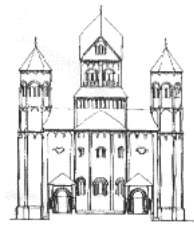
Jury still out on this one:

- CDF,D0: Decided against OODBMS
  - Transparent access without adequate resource management dangerous
- STAR,PHENIX: Explored Objectivity
  - Torre W: "STAR's initial persuit of Objectivity was a mistake"
- Compass: Event DB V1 ready; testing underway
- Babar: Objectivity up and running
  - Database performance satisfactory
  - Management and export issues remain
  - Bob J: OODBMS for event data "Not yet a proven concept"

Paris Sphicas CHEP2000



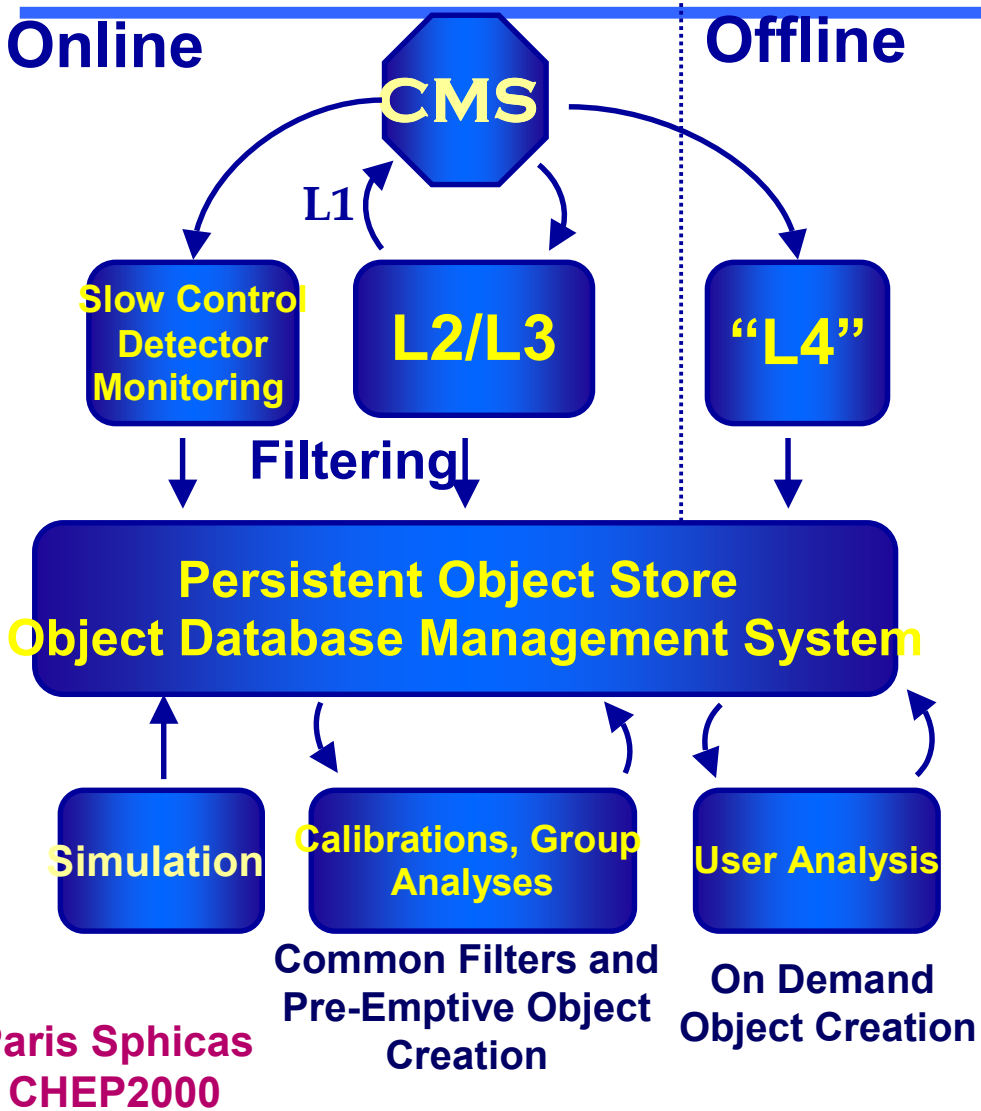
Data Storage/Access (II)



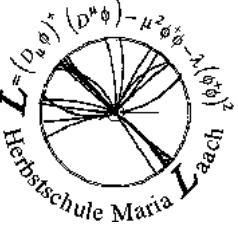
- There's more to data handling than a file format
  - Also need metadata (aka bookeeping); and optimization of resources (disks, tapes, robots, CPU)
  - Very large effort on "model", and thus, physical layout
    - Traditional: RAW, DST, μDST, pDST, NTUPLE
    - Maps onto physical layout: Shelf tape, Robot tape, disk, memory
- "Is transparent access on demand to all levels of hierarchy (a) necessary (b) Desirable (c) Possible"
  - For (a) and (b) no convincing argument for a positive answer
    - Thus, should we spend time on the feasibility?
  - Could be different at the LHC (?!)

Paris Sphicas
CHEP2000

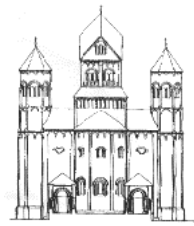
The Dream, part I



- From H. Newman:
  - "The DREAM" (on the left)
  - "Goal of location and medium transparency"
- From. V. White:
  - "DREAM -- minimum of work to store an object + DB provides query, security, integrity, backup, concurrency control, redundancy + has the performance of a hand-tuned object manager for your particular application"

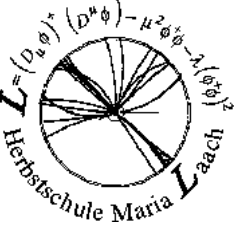


Data Storage/Access

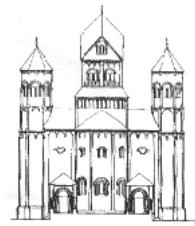


- The GRID: more support for the dream
  - promising data anywhere, anytime
  - Proponents say it's necessary because of the different scale
    - (much) more on this later
- Everybody wants to avoid data copying
  - Multiple claims that no-one intends to copy data.
  - In practice, we will, indeed, copy data
    - Hard to believe the 4-lepton samples will stay at CERN (only)
- THE question: are we (e.g. at the LHC) about to hit a phase transition?

Paris Sphicas
CHEP2000

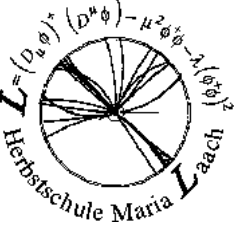


Is the LHC fundamentally different?

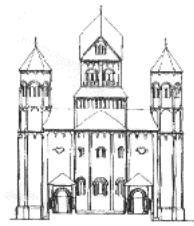


- LHC: a *natural* next step in progression of HEP needs
  - Current experiments off by factor 2-4 (only)
    - Compass: ~ 300 TB/year of RAW data
    - STAR: ~ 200 TB/year of RAW data
    - CDF: ~ 450 TB/year
  - Physics environment different, but if we can handle pileup, it's not drastically different
- LHC is (very) different in one aspect: *timescale*
  - We have the time to try more radical designs; even elegant, logical ones.
  - Thus, the Question: why not implement a phase transition in the mode of doing physics as well?

Paris Sphicas
CHEP2000

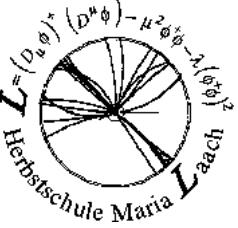


Changing for the LHC (?)

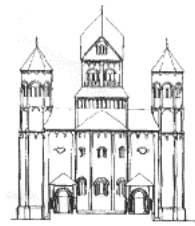


- Weak reason: not all is well in the land of OODBMS
  - See the Babar experience: "all is not sweetness and light"
    - Even some doubts regarding the true ultimate scaling: "Can the system keep up with billions of events and hundreds of physicists?"; "Our event store is not yet transparent: throughput problems, data distribution problems; still trying to get granularity right" B. Jacobsen
  - Can argue that these are not fundamental problems
- Stronger reason: because many expts will yield the answer on alternatives (e.g. the ROOT model)
  - Not using Objectivity does not mean we do not objectify
  - So yes, keep a close eye on what happens there

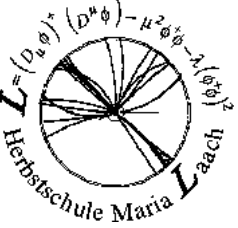
Paris Sphicas
CHEP2000



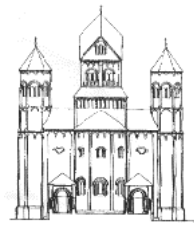
Changing for the LHC (?) (part II)



- Main reason: because *we have seen no proof that using an ODBMS lets us do something we cannot do using other means*
  - Yes we need metadata, queries, versioning
  - Does this mean we need a OO DBMS? (an elegant solution!)
  - Question ~ the same as “do we really need C++? We can do it all with FORTRAN”.
    - But this has been answered; and yes, OO can do things FORTRAN cannot



Open Source

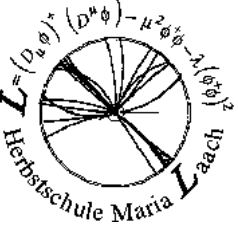


- A beautiful idea that *works* (unexpectedly)
- A few (necessary) observations:
  - People working on OS are *very young*

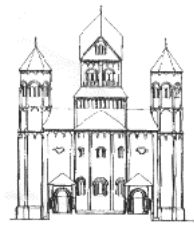
    (examples from the stars: Andreessen: 28; de Icaza: 26; Torvalds: 29)
  - People working on OS are *experts in computing*
    - They may be volunteers, but are working on computers, with computers, for a living. They are professionals.
  - People working on OS have an "unusual" culture/motives

    R. Stallman on de Icaza: "not only a capable software designer, but an idealistic and determined campaigner for computer users' freedom"
  - People working on OS are impressive
    - The world is watching; majority wants it; it will go on.

Paris Sphicas
CHEP2000

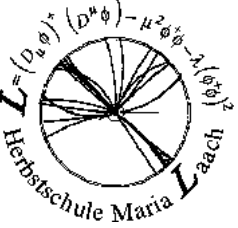


People in HEP

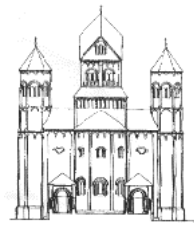


- A few reminders:
  - Average age higher (than in the Open Source group)
  - We are not experts in computing (as much as they are)
    - We learn QCD during the same period that people like Andreessen learn IPC calls and the client/server model
  - Our motivation is to do physics: understand Symmetry Breaking; study CP violation; meet gravity at the TeV scale
    - And "the system" rewards those who get there first. Recognition for a new technique (e.g. MWPC) is not very frequent.
  - People in HEP are impressive
    - In both good and bad ways
- The world is watching (counting $): we must not fail

Paris Sphicas
CHEP2000



Open Source in HEP



- Is there a future for Open Source in HEP?
  - Yes, there is, but not for everything in HEP
- OS (oversimplified but adequate) summary:

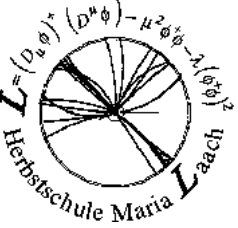
  1. "Write something good/useful, give the source to (capable) users and they will improve on it"

     (They'll even send you the improvements back, and you'll improve on those, and you'll release again, and they will use it and improve it further, and…)

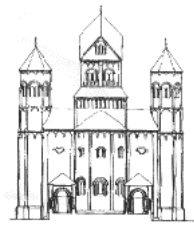
  2. "Adopt a good solution to a problem that has already been solved". Don't n-plicate work unnecessarily.

  3. "You earn respect for what you do, and only that; not for what you get appointed to do"

Paris Sphicas
CHEP2000

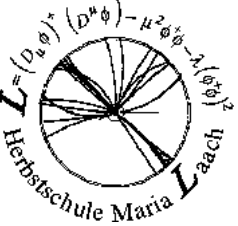


Can we find the people?

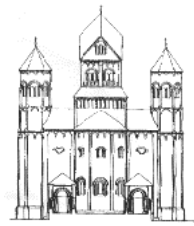


- In the broader HEP community we have people who fit these boundary conditions
  - They are not the average physicist
    - Either very young graduate students, or extremely bright, or computing professionals, or a combination
  - They do produce good/useful code (e.g. HYDRA)
    - And they play by the rules (and motivation) of hacker-stardom, not the rules of "publish or perish" or the rules of the 2000 person collaboration
  - We just need to rely on them, and *for some things, only on them*
    - It's time to recognize to leave some computing tasks to those who know computers better than we do

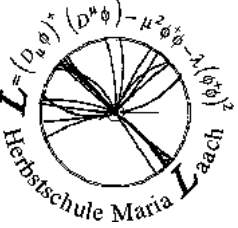
Paris Sphicas
CHEP2000



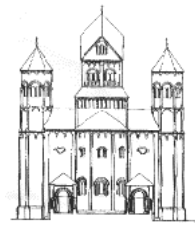
Open Source Model



- Small, efficient group
  - SAMBA: “15 people… 1/2 really active; 50% turnover on code”
  - Size is the same as the *core* team for the software of an experiment
    - Example (from T. Wenaus’ talk):
      - ~7 FTEs over 2 years in core offline
      - ~50 regular developers
      - ~70 regular users (140 total)
- Side conclusion: our teams can be as efficient as the OS teams, as long as they are staffed by the same kind/quality of people



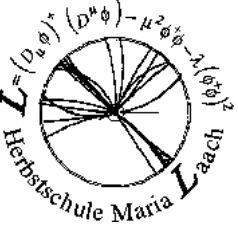
Back to the Open Source issue



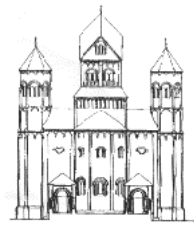
- Not everything that is currently produced is a good candidate for the OS model
  - Most kumacs should stay private. Because of quality reasons.
- Today's equivalent of OS in HEP: "common projects"
  - Why are CDF and D0 not using the same data model? V. White:

    "despite demonstrably similar requirements and overall access philosophy, 2 expts living in the same lab, encouragement from lab management for common solutions
    - CDF and D0 still have different hardware architectures and data access software implementations
  - There is no reason for the difference
- There are more things that can be solved "in common"

Paris Sphicas
CHEP2000

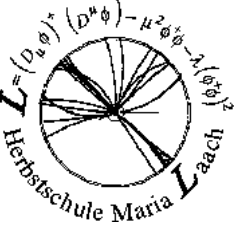


Final word on Open Source in HEP

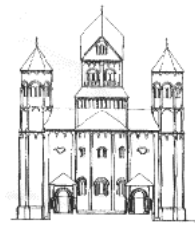


- We should adopt the model
  - Anyway, it already exists within HEP (ROOT project)
  - And it should be expanded; GEANT 4 seems like the natural candidate
    - GEANT x (x=3,4) is THE software product from HEP (web aside)
    - It's the ONLY standard product (outside PAW/ROOT) in HEP
    - It's already developed in a large collaboration (aka common project) fashion
  - We should expand on the idea; how about Joint CERN/DESY/KEK/FNAL/SLAC/university projects ?
    - 1-2 key people (I.e. experts) from each can work wonders
    - Logistics will be difficult; but all it takes is some willingness

Paris Sphicas
CHEP2000



Other Issues: OO (I)

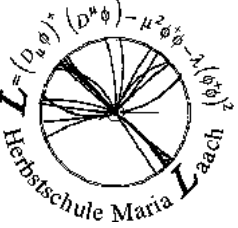


- OO methodology (C++, Java) is here to stay
  - All experiments reported near full to very high conversion factors from FORTRAN
    - All new students/postdocs/fellows know it (or want to learn it)
- OO methodology is not perfect; problems in deploying it

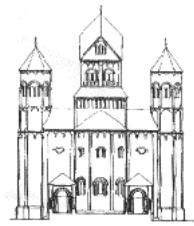
  D. Morrison: "OO … oversold ... as a computing panacea"; "occasional need for internal "public-relations""; "takes time and effort to "get it", to move beyond "F77++""

  B. Jacobsen: "C++ is a pig of a language from a memory leak point of view"; "much existing expertise of doubtful applicability"; "C++ advocates had limited design experience"; "Mismatch between enthusiasm and effectiveness"

  M. Shapiro: "Bad C++ is worse than bad FORTRAN"; "memory management an issue — constant battle with memory leaks"



Other Issues: OO (II)



- Main reasons for the rush to OO have been:
  - Best way (known) to write a 10Mline program
  - Best way (known) to maintain a 10Mline program for 10 years
- Guinea pigs agree: OO has delivered on these fronts

  Morrison: "make big computing problem tractable…"

  Meritt/Shapiro: "Yes, we have successfully built large C++ systems"
    - CDF: 1.3 million lines of code; DØ: 285 cvs package

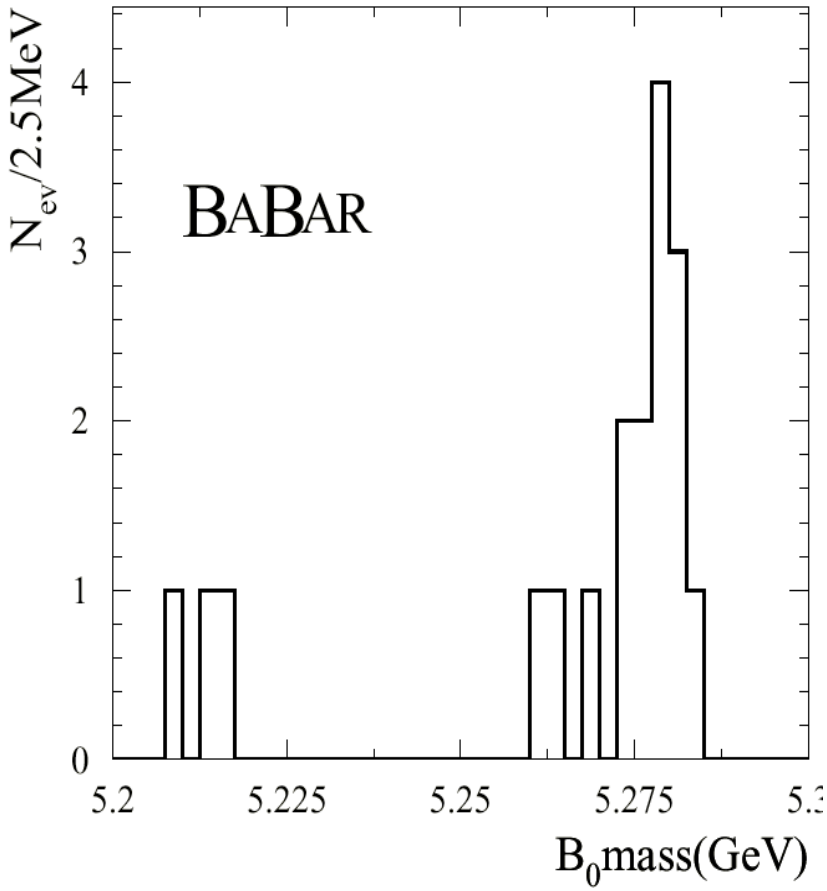
  "Yes, we are building data handling systems that approach LHC sizes"
    - 0.75 - 1.0 PB storage capacity (per exp't) will be available

  "Will the larger community find them highly usable or barely usable?"
  - (My answer) yes, if supplemented with the right PAW-like product

Other Issues: OO (IV)

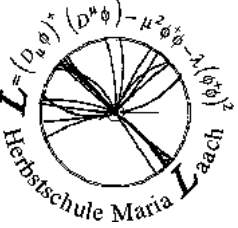
- Above all, it works on the field
- Also important:
  - No-one reported an intent to go back to FORTRAN
  - No-one expressed any longing for the "good old FORTRAN days"

    M. Shapiro: "all expts agree that C++ is the righ choice…"
- Conclusion: if you want to write software, learn OO

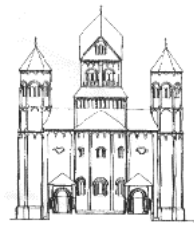


$B \rightarrow J/\psi K_s$

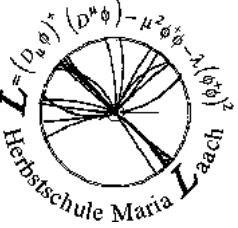
Paris Sphicas
CHEP2000



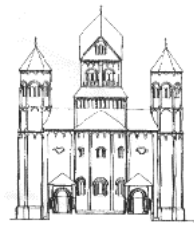
Other issues: The Dream, part II



- Vulgarization of the dream for non-experts:
  - That doing physics will be easy, really easy
- Design team reading URD:
  - Define “doing physics”, define “(really) easy”
  - Is the concept of doing everything off of a ODBMS enough to satisfy “the dream”?
- HEPhysicist:
  - Well, no. Previous transparency is a SDD, not a URD
    - So, improve on URD.

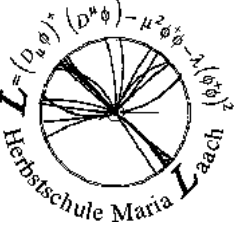


Towards a URD for the dream

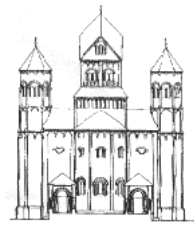


- "Doing physics" includes:
  - Lots of obvious things
    - Calibrated data; Small data sets (for HUMAN not CPU reasons); Easy access to data (networks, etc); easy language to tell computer what to do, etc etc etc
  - Ability to play with Data and Monte Carlo
    - See what happens when one relaxes/tightens "cuts"
    - Check if a new data set behaves the same way as an older one
    - See (quickly, i.e. few days) how GMSB vs AMSB differ in "signatures"
    - …
  - Ability to involve the maximum # of physicists on the expt
    - B. Jacobsen: "Can senior people with good intuition contribute?"
      - We should make sure they can

Paris Sphicas
CHEP2000

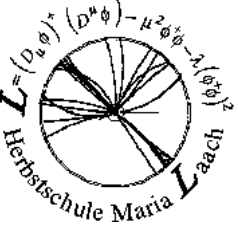


The Physics Analysis Workstation

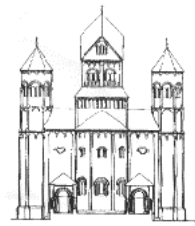


- It brought physics analysis to the masses
  - Its impact on our daily work equivalent to that of
    - The spreadsheet (e.g. EXCEL) in accounting
    - The Web in acquiring information on anything, e.g. Padova
  - It was (and still is) easy to learn and use
    - "NTUPLE" became a word that was used by essentially all "senior people with good intuition"
  - And (perhaps above all) it is "interactive"
    - Interactive :== [T(answer)–T(question) = O(sec/min)]
    - Just like EXCEL and the Web
- Need imrpovements here:
  - OO, unified access to high level physics objects,

Paris Sphicas
CHEP2000



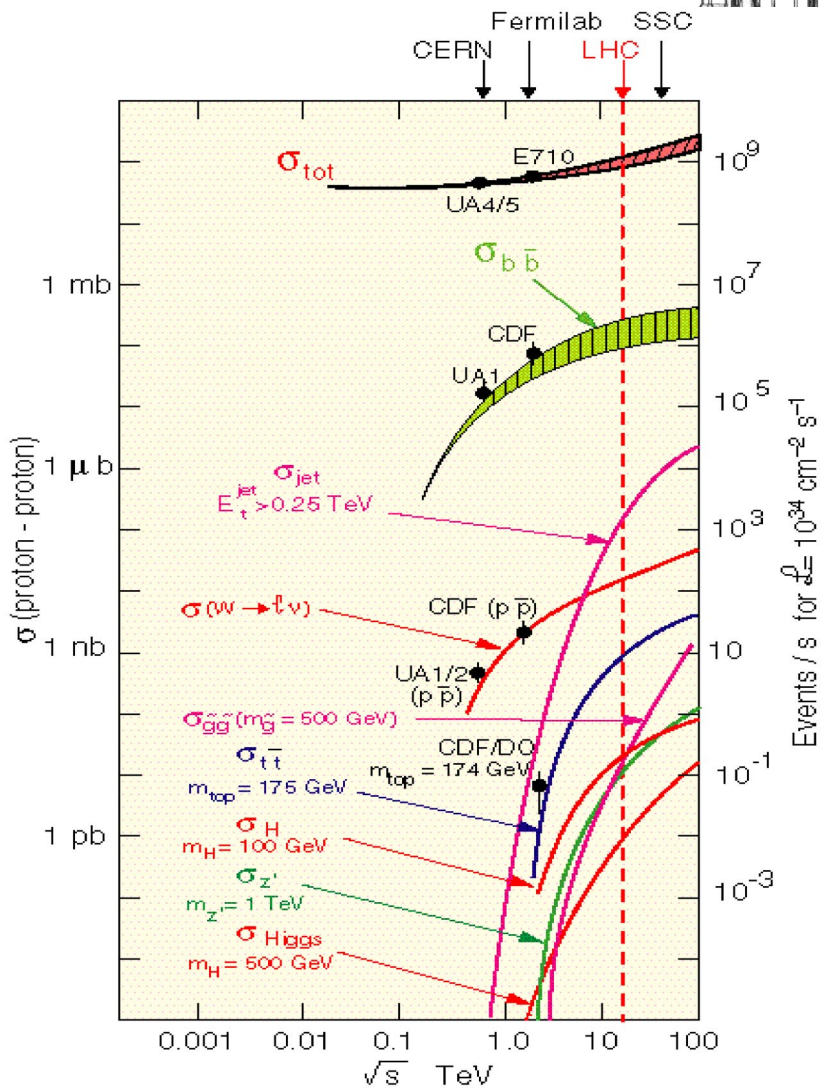
Need more PAW-like capability



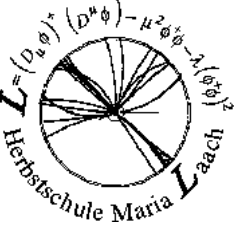
- Reasons for wanting more:
  - First, it has to be OO
  - Second, more integration with other components of our analysis environment
    - Why not click on a track and get the event display with the track highlighted?
  - Third, like all products go, PAW can take some improvements
- But before that, we need a model for accessing our high-level physics objects
  - Do we keep the full objects and read them in as such?
    - Do we store a "secondary vertex" in the "NTUPLEs"?
- This issue is also worth wondering about, now...

Other things to worry about (I)

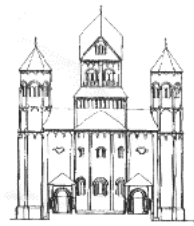
- Cross sections for various physics processes vary over many orders of magnitude; e.g. at LHC:
  - Bunch crossing: 4×10^6 Hz
  - $W \rightarrow \ell \nu$: 10^2 Hz
  - Higgs (600 GeV/c^2): 0.01 Hz
- Selection (100 Hz storage):
  - Online: $1:4 \times 10^4$;
  - Offline: $1:10^4$
- Must monitor the selection



Paris Sphicas
CHEP2000



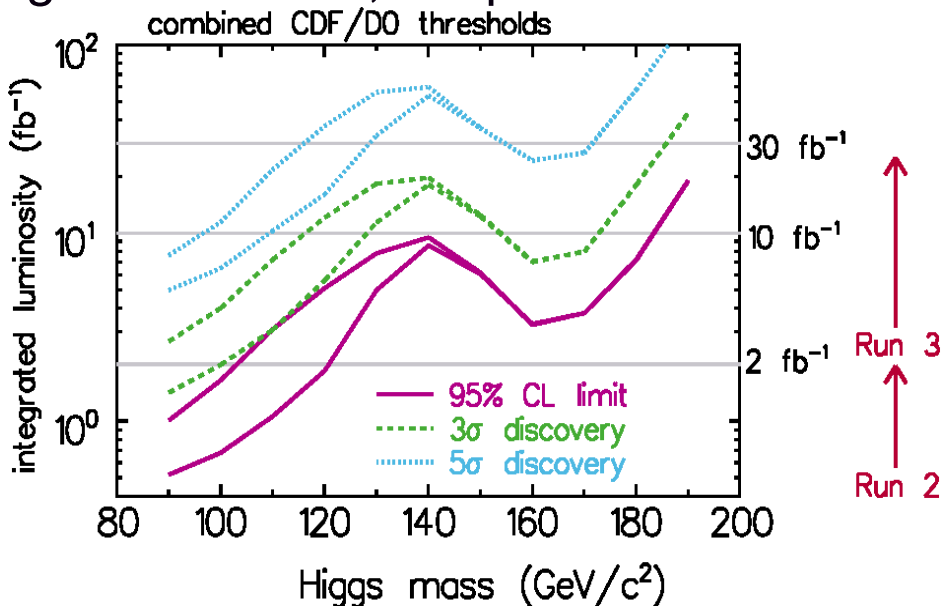
What we should be talking about



- How we will perform these fine selections
  - Level-1, Level-2, Level-3, Offline, “PAW”, etc.
- How we will monitor them
  - Level-1, Level-2, Level-3, Offline, “PAW”, etc.
- What we will do in order not to regret them
  - Level-1, Level-2, Level-3, Offline, “PAW”, etc.
- What new algorithms we need to do physics when working in successive approximations
  - Have we really ran out of new techniques and algorithms?
  - No, we just need time to absorb more advanced (e.g. mathematical) techniques

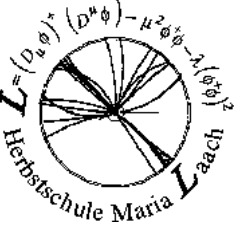
Algorithms, reconstruction, analysis

- Basic HEP analysis uses mostly kinematics
  - Three- and four-vector manipulations
  - Some new techniques, e.g. Neural Nets, adopted
    - But still "suspect" (!)
- Complete lack of follow-up on new techniques
  - ICA, genetic algorithms

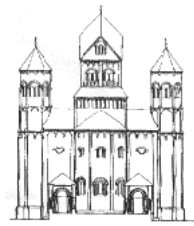


- Because instead, we spend our time on things we are not so good at

Paris Sphicas
CHEP2000



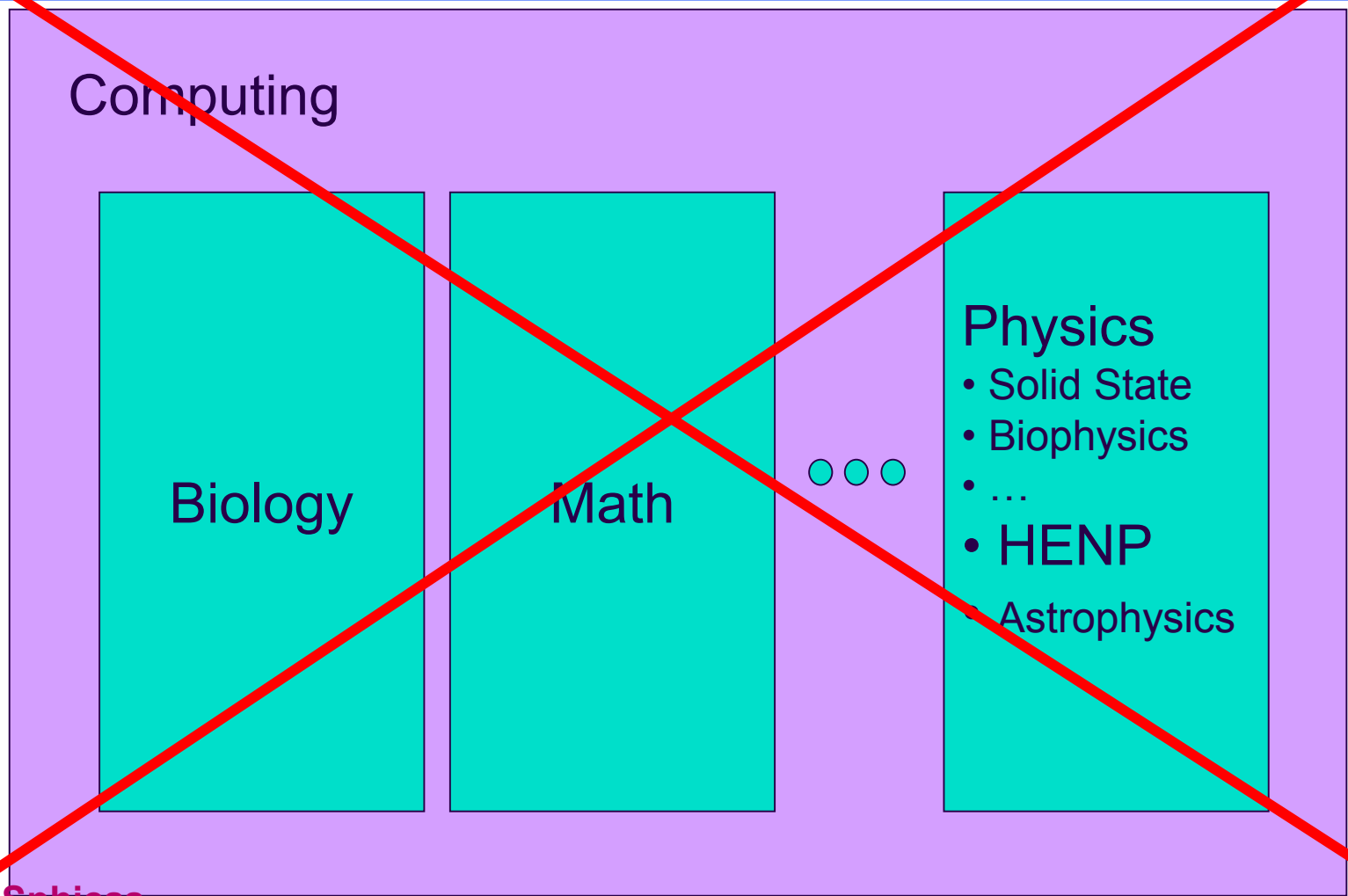
Relying on experts



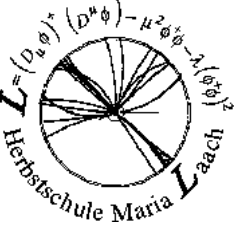
- In some cases we are trying to play “computer scientist”
  - We shouldn’t. We should leave this task to computer scientists, i.e. professionals. At least for the core software.
- We have done that already with the big detectors
  - I would not work on an experiment where the mechanics of the magnet is designed by a “jack of all trades” HEPhysicist who learned it on the job.
    - Unless the HEPhysicist was a uniquely gifted person
  - Complexity (detector and computing) has overtaken the average HEPhysicist
    - Engineers are now necessary; we can work with them; guide them; help them; disagree with them

Paris Sphicas
CHEP2000

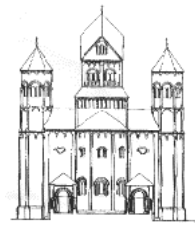
High Energy Physics in Computing



Paris Sphicas
CHEP2000



Computing in HENP



High Energy and Nuclear Physics

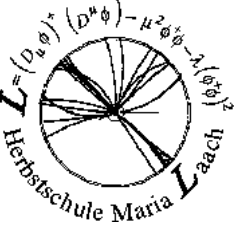
Tracking

Calorimetry

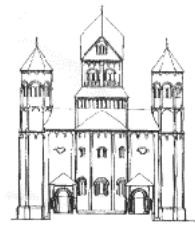
○○○

Computing

Paris Sphicas
CHEP2000

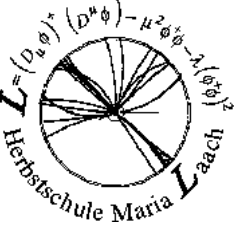


Conclusion

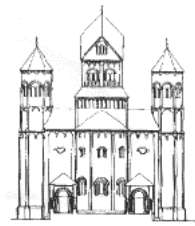


- Data access: ODBMS (so far) not proven
- Open Source: a blessing, in the right hands
- OO: it works (and delivers on large projects)
  - And it's here to stay
- Miscellanea:
  - Computing is a science on its own; it's not trivial
    - Make more use of computing professionals
  - Concentrate on what we know best:
    - Spend more time in defining/helping end-user analysis (PAW)
    - Control and Monitor the incredible selection
    - Learn how to do more "computation" — and use it.

Paris Sphicas
CHEP2000

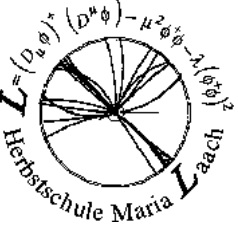


Conclusions

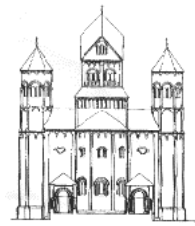


- If you have a lot of data to manage and access today you must
  - think carefully about how you store it, how you wish to access it, and how you will control access to it
  - be aware of media costs
  - design a system for robustness and uptime (especially if you use relatively inexpensive tape media)
  - design a system for active and managed access to all hierarchies of storage - disk, tape in robot & tape on shelf
- For the next generation of experiments
  - we hope for better network bandwidth and a truly distributed system
  - we investigate OO databases for their potential to provide random access to sub-parts of event data

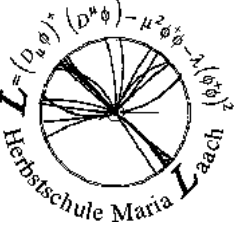
Paris Sphicas
CHEP2000



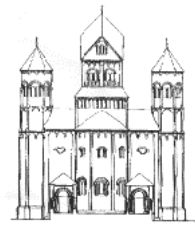
Role of computer networking (1)



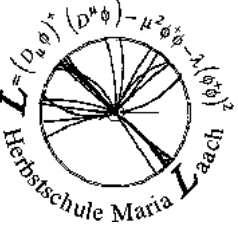
- State-of-the-art computer networking enables large international collaborations
  - needed for all aspects of collaborative work
    - to write the proposal,
    - produce and agree on the designs of the components and systems,
    - collaborate on overall planning and integration of the detector, confer on all aspects of the device, including the final physics results, and
    - provide information to collaborators and to the physics community and general public
  - Data from the experiment lives more-and-more on the network
    - All levels: raw, dst, aod, ntuple, draft-paper, paper



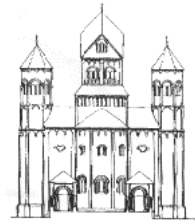
Role of computer networking (2)

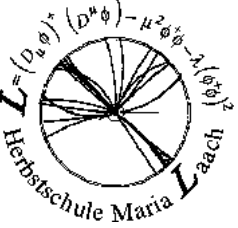


- HEP developed its own national network in the early 1980s
- National research network backbones generally provide adequate support to HEP and other sciences.
- Specific network connections are used where HEP has found it necessary to support special capabilities that could not be supplied efficiently or capably enough through more general networks.
  - US-CERN, several HEP links in Europe…
- Dedicated HEP links are needed in special cases because
  - HEP requirements can be large and can overwhelm those of researchers in other fields
  - because regional networks do not give top priority to interregional connections

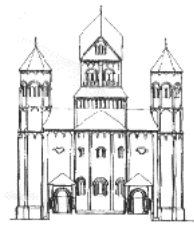


Specialized computing systems

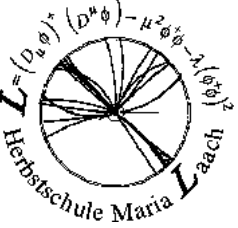




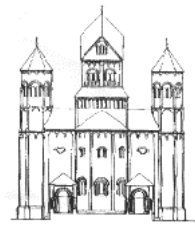
Lattice QCD



- Lattice QCD is a powerful approach to study QCD and to calculate fundamental quantities in the theory.
- Lattice QCD calculations require extremely large computing power in a tightly-coupled computing architecture (because of the demand for fast, low-latency communications).
- The computers used for this are almost always special-purpose machines, designed for this class of calculations
- Examples include, APE, Columbia machine, ACPMAPS, etc.

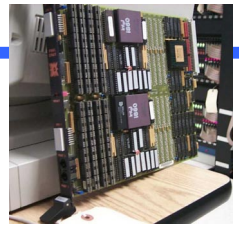


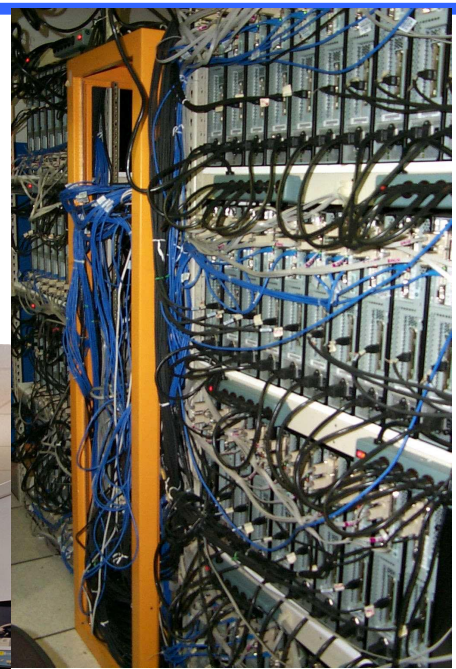
Commodity Lattice Gauge Machine

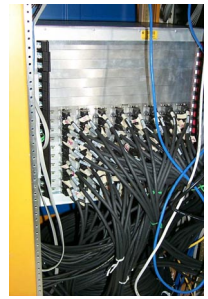


- Idea: Take advantage of commodity hardware and software to build a large lattice QCD machine.
- Goal: 10 Teraflop peak performance as cheaply as possible.
- R&D at Fermilab (similar work at Jefferson lab).
  - Small Machine (80 dual PC's) has been purchased and is being integrated at Fermilab.
  - Much larger machine (on order 1000 PC's) will be built assuming that funding is available, no serious problems are found in scaling, etc.
  - Workshop was held March 26-28 at Fermilab to discuss the current ideas and progress.

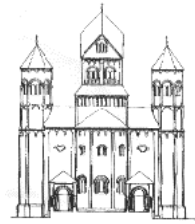
Old and New Lattice Gauge Computing at Fermilab



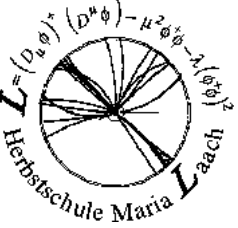




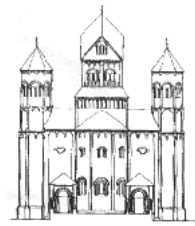
Many more technical questions to answer (1)



- Operating system:
  - UNIX seems to be favored for data handling and analysis,
  - LINUX is most cost effective
- Mainframe vs. commodity computing:
  - commodity computing can provide many solutions
  - Only affordable solution for future requirements
  - How to operate several thousand nodes?
  - How to write applications to benefit from several thousand nodes?
- Data access and formats:
  - Metadata databases, event storage

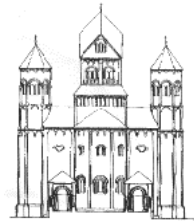


Many more technical questions to answer (2)



- Commercial vs. custom software, public domain
- Programming languages:
  - Compiled languages for CPU intensive parts
  - Scripting languages provide excellent frameworks
- How to handle and control big numbers in big detectors:
  - Number of channels, modules improves (several millions of channels, hundreds of modules
  - Need new automatic tools to calibrate, monitor and align channels

Some more thoughts



- Computing for HEP experiments is costly
  - In $$'s, people and time
  - Need R&D, prototyping and test-beds to develop solutions and validate choices
- Improving the engineering aspect of computing for HEP experiments is essential
  - Treat computing and software as a project (see www.pmi.org):
    - Project lifecycles, milestones, resource estimates, reviews
- Documenting conditions and work performed is essential for success
  - Track detector building for 20 years
  - Log data taking and processing conditions
  - Analysis steps, algorithms, cuts

As transparent and automatic as possible

User View of PVDG Architecture